Centro de Investigação Operacional

An Interactive Approach for Bicriteria Transportation Problems

Inês Marques, Maria Eugénia Captivo and João Clímaco

CIO – Working Paper 2/2007

An Interactive Approach for Bicriteria Transportation Problems

Inês Marques, M. Eugénia Captivo
Universidade de Lisboa, Faculdade de Ciências, Centro de Investigação Operacional
DEIO, Edifício C6 - Piso 4, Campo Grande, 1749-016 Lisboa, Portugal
ines.marques@fc.ul.pt, mecaptivo@fc.ul.pt

and

João Clímaco
Universidade de Coimbra, Faculdade de Economia, INESC Coimbra
Av. Dias da Silva, 165, 3004-512 Coimbra, Portugal
jclimaco@inescc.pt

March 9, 2007

Abstract

An interactive approach for Bicriteria Transportation Problems, allowing for non-integer solutions, is presented.

The decision-maker (DM) conducts the search by introducing upper bounds on the values of one or both criteria, adding one/two additional constraints to the original problem structure. So, usual algorithms for solving transportation problems cannot be used.

Two extensions of the network simplex method were implemented. One to support two additional restrictions, and the other for one additional restriction only. Both revealed to be less efficient than Cplex software, however, solving higher dimension problems.

The developed algorithms were introduced in a Decision Support System.

Keywords: Transportation Problems, Bicriteria Programming, Decision Support Systems, Network Simplex.

1 Introduction

In this work we study an interactive approach for bicriteria transportation problems. These problems can occur when, in the presence of a problem with the classical transportation structure, there are two distinct and conflicting criteria orienting the decision. For instance, in the case of transporting noxious goods it is relevant to minimize the risk of transportation while minimizing its total cost. Still further in the transportation of perishable goods the minimization of the total cost may be conflicting with another objective of minimizing the transportation time.

We consider the classic transportation problem with two linear objective functions. The criteria that can be considered are, for instance, the minimization of the total transportation cost and its duration, or the minimization of the total deterioration of goods (relevant for certain perishable goods) and the use of certain scarce resource such as energy, the maximization of safety, reliability, etc.

These problems may be applied to oil companies that need to transport fuel from refineries to gas stations. The aim is to minimize the transportation costs, as well as, to maximize the security.

If the objectives are conflicting there is no more an optimal solution. Alternatively, there is a set of efficient solutions which are those where the improvement in one objective can only be made at the expense of the other objective. The large number of efficient solutions implies a decision process, for which we intend to contribute with a decision support system.

A bicriteria linear problem (with only supported efficient solutions) and a bicriteria integer problem (with supported and non-supported efficient solutions) have different efficient solution structures. Therefore, in bicriteria programming it is necessary to consider explicitly the nature of the decision variables [7].

The total unimodularity of a transportation problem constraint matrix only guarantees the Pareto frontier extreme points to be integer solutions. In this work a linear problem is considered, thus the decision variables can take continuous values and the efficient solutions are all in the Pareto frontier.

So, the model considered can have the following formulation:

$$\begin{array}{ll} \min & z_1(x) = c^1 x \\ \min & z_2(x) = c^2 x \\ s.t. & \quad x \in X \end{array}$$

with c^1 and c^2 being the cost coefficient vectors of objective function one and two, respectively, and X being the feasible set of a transportation problem ($X = \{Ax = b,\ x \geq 0\}$, A being a node-arc incidence matrix and b the supply and demand vector).

There are, in the literature, some a priori, a posteriori and interactive approaches to this model. In an a priori approach, only one solution is determined according to the preferences expressed by the decision-maker (DM) in the beginning of the decision process. For this kind of approach, the DM must have sufficient knowledge about the problem to believe and express some preferences. Here, we emphasize a goal programming approach by Lee and Moore [13], an utility function optimization by Diaz [5] and a lexicographic optimization by Stewart and Ittmann [15].

On the other hand, a posteriori approaches present to the DM a set of solutions and he selects the most preferred one. In this situation, the information that is given to the DM can be very intensive without contributing to an effective knowledge of the problem by the DM. Additionally, it will lead to an unnecessary computational effort. A posteriori approaches to the bicriteria transportation problem are due to a weighted sum method from Aneja and Nair [1], a duality study by Isermann [12], an adjacency search by Diaz [6] and a dynamic programming from Gallagher and Saleh [9].

In interactive approaches, the solutions are explored by a progressive articulation of the DM's preferences. This enables to reduce the computational effort of the calculation process and to aid the DM in the decision process by providing a progressive learning of the problem. As far as we know there is only one interactive approach to bicriteria transportation problems being an adjacency search by Ringuest and Rinks [14]. There is still further an interactive package called TRIMAP that is dedicated to solving three-objective linear problems and is illustrated by Clímaco *et al.* [4] with a transportation problem.

In the literature it is not clear the nature of the decision variables regarding the respective efficient solutions. In none of these works the non-supported efficient solutions were determined. Whereas to the supported solutions only the extreme points were determined. Moreover, if in some cases the linear transportation problem is explicitly indicated, in others it seems that the integer problem is solved solely through linear relaxation. The exception is the work by Clímaco *et al.* [4].

In the following we go through the interactive algorithm used (section 2) and the two approaches that were undertaken, the problems those approaches originate and the algorithm developed to solve those resulting problems (section 3). In section 4 we present some computational experience. We will end in section 5 by giving a short picture of the decision support system that was developed.

With this work we intend to give a contribute in three ways: in the studying of bicriteria transportation problems; providing an efficient algorithm for solving transportation problems with additional linear constraints (section 3); and providing also a practical tool to aid the decision in the context presented.

In what follows we assume that the reader is familiar with transportation problems and the network simplex method. For the required background on these we suggest Bazaraa *et al.* [2].

2 Interactive Algorithm

We developed an interactive approach to aid the decision process within a bicriteria transportation problem. In this kind of approach there is an interaction with the decision-maker (DM) and he conducts the search as an oriented process.

The interactive algorithm starts by calculating the lexicographic minima. These are the efficient solutions corresponding to the individual optimum of each criteria. If the two solutions are the same, then there is no conflict between the two criteria and we have an optimal solution to the problem. Otherwise, their objective values indicate the space of variation for the value of the two criteria in the problem efficient region. If $z^{1*} = (z_1^{1*}, z_2^{1*})$ and $z^{2*} = (z_1^{2*}, z_2^{2*})$ are the lexicographic minima for the first and the second criteria then, in the efficient region, the first criteria will take values between z_1^{1*} and z_1^{2*} and the second one between z_2^{2*} and z_2^{1*}.

After this initial step, dialogue phases with the DM interchanges with computation phases until he considers having sufficient information about the problem allowing him to make the decision. In the dialogue phase, the DM analyze the solutions provided and defines the next region to pursue the search. In the computation phase, the indications provided by the DM are added to the problem and the resulting problems are solved.

We considered two distinct approaches for the way that the DM indicates the next region to pursue the search, providing two practical alternatives to the DM in the decision support system. Both determine efficient solutions.

2.1 Approach 1

In the first approach we allow the DM to focalize the search for efficient solutions by introducing upper bounds on the value of both objective functions. This can be in the form of a pair of non-dominated solutions or on upper bound values directly on the value of both criteria. It follows that two linear constraints are added to the original transportation problem structure.

To determine efficient solutions we use in this approach the weighted sum method, and thus the objective functions are convex combined in only one criteria. It is well known that, when the decision variables can take continuous values, the linearity of the constraints and objectives makes it possible to find all efficient solutions by varying the weighting parameter value in $]0, 1[$ [7].

Thus, the resulting problem of this approach is $\min\{cx : \ x \in X, \ c^1x \leq \alpha_1, \ c^2x \leq \alpha_2\}, \ c = \lambda c^1 + (1 - \lambda)c^2$ and $0 < \lambda < 1$.

2.2 Approach 2

Alternatively, the second approach allows the introduction of an upper bound in the value of only one objective function. This follows that one linear constraint is added to the original transportation problem structure corresponding to the objective function that is bounded by the DM.

For this approach we used the α-constraint method, where the objective function that is bounded by the DM is transformed into a constraint and the other original objective function is optimized. This method allows the determination of whatever efficient solution [7].

The resulting problems of this approach can be either $\min\{c^1x : \ x \in X, \ c^2x \leq \alpha_2\}$ or $\min\{c^2x : x \in X, \ c^1x \leq \alpha_1\}$.

The efficient solution is really obtained in the equality of the additional restriction. This is due to the efficient solutions structure and the decision variables continuous domain. Thus, this approach allows the DM to know an efficient solution with a specific value in one criterion.

2.3 Resulting Problems

Accordingly, the resulting problems of both approaches consist of simple transportation problems with additional linear constraints and only one objective function. With the additional constraints, usual algorithms for solving transportation problems cannot be used.

Our aim is to develop an algorithm capable to solve both resulting problems based on the original transportation structure (to be introduced in the decision support system) and understand if its performance becomes more efficient than the resolution with a general solver.

Therefore, we extended the network simplex method to support either two additional restrictions or only one restriction based on Chen and Saigal [3] work for the minimum cost network flow problem. The algorithm developed is also valid for any number of additional constraints to the transportation structure.

The algorithm developed is based on the partitioned form of the basis matrix and this allows to make use of the procedures from the network simplex method.

2.3.1 Basis Matrix Characterization

Lets consider the matricial formulation that emerges from the first approach.

$$\min_x\{cx : Ax = b, \ Cx + Ps = \alpha, \ x \geq 0, \ s \geq 0\}$$

A is a node-arc incidence matrix and slack variables s are introduced to transform the additional restrictions to equality.

Adding the artificial variable/root arc e_j to complete the rank of the constraint matrix, we obtain

$$\overline{A} = \left[\begin{array}{ccc} A & & e_j \\ C & P & \end{array} \right] \begin{array}{l} \} \ m+n \\ \} \ 2 \end{array}$$

So, a basis for $\overline{A}$ can be

$$\overline{B} = \left[\begin{array}{cc} B & D \\ C^B & F \end{array} \right]$$

B is a subset of the full node-arc incidence matrix ($[A|e_j]$) and corresponds to the basis of the network structure, so it can be represented by a rooted spanning tree. C^B corresponds to a cost matrix and its elements are the cost components in the two objective functions of the basic variables in the tree. F is the basis component corresponding to the additional structure and has, in this case, two basic variables associated that can either be arc variables or slack variables. D is a subset of the node-arc incidence matrix corresponding to the two additional basic variables. A slack basic variable corresponds to a column of zero elements in D.

So, we have $m+n$ basic variables in the rooted spanning tree (corresponding to the network structure) and 2 additional basic variables (corresponding to the additional constraints).

Note that when the slack variables are basic variables they can only belong to the basis additional component as they are not represented by any arc in the network.

The partitioned basis matrix is the base of the network simplex extension to support additional constraints. This extension is also valid for any number of additional restrictions. Nevertheless, it becomes much more efficient when the additional structure reduces to a small part when compared to the network portion of the structural constraints.

3 Network Simplex Method for Transportation Problems with Additional Constraints

In this section we present the algorithm developed to solve the transportation problems with additional constraints, resulting from the interactive approaches for the bicriteria transportation problem. In the subsection below we will try to illustrate each one of the algorithm steps with a simple example.

Before going through the algorithm we introduce some useful notation.

We will call TP to $\min\{cx : \ x \in X\}$, TPg to $\min\{c^g x : \ x \in X\}$, $g = 1, 2$, and TP++ to $\min\{cx : \ x \in X, \ c^1 x \leq \alpha_1, \ c^2 x \leq \alpha_2\}$.

Generally, an index superscript 1 refers to the network structure and 2 to the additional structure. So, x^2 refers to the additional basic variables in TP++ and x_h^2 $(h = 1, 2)$ specifies the first or second additional basic variable. Accordingly, Π^1 is the vector containing the dual variables related to the network constraints and Π^2 is the vector containing the dual variables associated with the additional constraints. Updated column y of the entering variable is also partitioned in y^1 and y^2 according the basis matrix partition.

Let $\delta_{h(i)}$ and $\delta_{h(j)}$ be the dual variables for the extreme nodes i and j of the additional basic variable x_h^2 in TP, and $\delta_{h(i)}^g$ and $\delta_{h(j)}^g$ the dual variables for the same nodes in TPg $(g = 1, 2)$. If x_h^2 is a slack variable then $\delta_{h(i)} = \delta_{h(j)} = 0$ and $\delta_{h(i)}^g = \delta_{h(j)}^g = 0, \ g = 1, 2$. Finally, let σ_h be the reduced cost of the variable x_h^2 in TP and $\overline{c}_{x_h^2}^{\,g}$ the reduced cost of x_h^2 in TPg $(g = 1, 2)$.

If $\overline{B}$ is the basis matrix then its inverse is

$$\overline{B}^{-1} = \left[\begin{array}{cc} B^{-1} + B^{-1}DQ^{-1}C^{B}B^{-1} & -B^{-1}DQ^{-1} \\ -Q^{-1}C^{B}B^{-1} & Q^{-1} \end{array} \right]$$

with $Q = F - C^{B}B^{-1}D$. The auxiliary matrix Q supports the additional component of the inverse basis matrix.

As we are talking about a typical simplex algorithm, the network simplex method for transportation problems with additional constraints follows the same structure:

1. Determine an initial basic feasible solution

   For the first approach we used the two-phase method; for the second approach we took advantage of a basic feasible solution already calculated.

   Calculate $\delta^{g}_{h(i)}$ and $\delta^{g}_{h(j)}$ ($h = 1, 2$ and $g = 1, 2$), using the procedure for the dual calculations from the network simplex algorithm.

   Construct the auxiliary matrix $Q_{2\times 2}$: $Q_{gh} = \overline{c}^{\,g}_{x^2_h} = c^{g}_{x^2_h} - \delta^{g}_{h(i)} + \delta^{g}_{h(j)}$ ($g = 1, 2$ and $h = 1, 2$).

   Invert Q.

2. Calculate the values of the dual variables

   Determine $\delta_{h(i)}$ and $\delta_{h(j)}$ using the procedure for the dual calculations from the network simplex method.

   Calculate $\sigma_h = c_{x^2_h} - \delta_{h(i)} + \delta_{h(j)},\ h = 1, 2$.

   The dual variables of the additional constraints are given by $\pi^2_h = \sigma_1 Q^{-1}_{1h} + \sigma_2 Q^{-1}_{2h}\ ,\ h = 1, 2$.

   Calculate the parameters $d_{ij} = c_{ij} - c^1_{ij}\pi^2_1 - c^2_{ij}\pi^2_2$ for (i, j) in the spanning tree.

   Π^1 is determined using the procedure for the dual calculations from the network simplex method based on d_{ij} ($\pi^1_i\ ,\ i \in I$ and $\pi^1_j\ ,\ j \in J :\ d_{ij} - \pi^1_i + \pi^1_j = 0\ ,\ (i, j) \in B$).

3. Calculate the value of the reduced costs and test optimality

   The value of the reduced costs are given by

   $\overline{c}_{ij} = c_{ij} - \pi^1_i + \pi^1_j - c^1_{ij}\pi^2_1 - c^2_{ij}\pi^2_2$, for the arc variables (i, j) $(i \in I, j \in J)$ and

   $\overline{c}_{s_r} = -\pi^2_r$, for the slack variables $(r = 1, 2)$.

   If $\overline{c}_{st} = \min\ \{\overline{c}_{ij},\ i \in I, j \in J\ ;\ \overline{c}_{s_r},\ r = 1, 2\} \geq 0$ then the current solution is optimal for TP++ and the algorithm stops; otherwise x_{st} is selected to enter the basis.

4. Update the column corresponding to the entering variable

   Initialize y^1 with the cycle trace procedure from the network simplex algorithm for x_{st}. If x_{st} is a slack variable then y^1 is a vector of null elements.

   Calculate $\Delta_g = c^g_{st} - \sum_{(i,j):\ x_{ij}>0}\ y^1_{ij}\ c^g_{ij}\ ,\ g = 1, 2$.

   The part of the column that relates to the additional component is given by $y^2_h = Q^{-1}_{h1}\Delta_1 + Q^{-1}_{h2}\Delta_2\ ,\ h = 1, 2$.

   Update y^1 with two running of the cycle trace procedure from the network simplex algorithm, one for each additional basic variable, adding from the old values the following ones: $-y^2_1$ and

$-y_2^2$ in the place of $+1$, and y_1^2 and y_2^2 in the place of -1, for the respective additional basic variable. If the additional variable x_h^2 is a slack variable then the cycle trace for this variable does not proceed.

5. Choose the variable to leave the basis

   The variable leaving the basis, x_{ql}, is the one that defines the flow variation given by $\Delta = \min\{\frac{x_{ij}}{y_{ij}^1} : \ y_{ij}^1 > 0, i \in I, j \in J \ ; \ \frac{x_h^2}{y_h^2} : \ y_h^2 > 0, h = 1, 2\}$.

6. Update the solution

   Update the values of each variable according to

   $x_{ij}^1 \ \leftarrow \ x_{ij}^1 - y_{ij}^1\Delta$, for the arcs (i, j) in the spanning tree,

   $x_{st} = \Delta$ and

   $x_h^2 \ \leftarrow \ x_h^2 - y_h^2\Delta$, for the additional basic variables $(h = 1, 2)$.

7. Update the basis and return to 2

   For the basis update there are two possible procedures depending on which component of the basis the variable leaving the base belongs to: on the first case we assume that it belongs to the additional component and on the second one that it belongs to the spanning tree.

   (a) The variable leaving the base, x_{ql}, is an additional basic variable.

   The basis update is performed outside the spanning tree so only the inverse of the Q matrix is updated and this is done using the standard pivoting procedure from the simplex algorithm.

   (b) The variable leaving the base, x_{ql}, belongs to the spanning tree.

   Test if the variable leaving the base is on a cycle formed with one of the additional basic variables. Two cases can happen:

   i. If there is no cycle containing x_{ql} and one x_h^2 $(h = 1, 2)$
   The basis update involves only the tree and so the tree update corresponds to the update basis procedure from the network simplex algorithm.

   ii. Otherwise,
   The basis update involves the two component of the basis and it is performed in two steps.
   In the first step the leaving variable changes with the additional basic variable with which it forms a cycle. This involves the update basis procedure from the network simplex algorithm and the following matrix operations:
   $Q_{h1}^{-1} \ \leftarrow \ -\mu_1 Q_{11}^{-1} - \mu_2 Q_{21}^{-1}$ and $Q_{h2}^{-1} \ \leftarrow \ -\mu_1 Q_{12}^{-1} - \mu_2 Q_{22}^{-1}$ with $h = 1$ if the exchange is performed with the first additional basic variable and $h = 2$ if with the second one; and

$$\mu_r = \begin{cases} 1, & \text{if } x_{ql} \text{ and the additional basic variable } r \text{ have the same direction in the cycle} \\ -1, & \text{if } x_{ql} \text{ and the additional basic variable } r \text{ have opposite directions in the cycle} \\ 0, & \text{if } x_{ql} \text{ and the additional basic variable } r \text{ do not form a cycle} \end{cases}$$

   After this step, the leaving variable is an additional basic variable and we are back to the first hypothesis, proceeding with the standard pivoting from the simplex algorithm to update the inverse of the Q matrix.

The scheme of the algorithm is presented in figure 1.

The presentation of the algorithm was based on the resulting problem from the first approach, so for transportation problems with two additional constraints. Nevertheless, this can be used for whatever number of additional constraints.

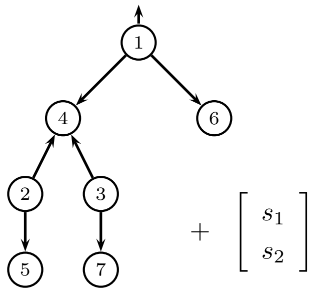
3.1 Illustration

Consider the following instance of a bicriteria transportation problem:

$$c^1 = \begin{bmatrix} 6 & 1 & 2 & 5 \\ 3 & 5 & 4 & 1 \\ 5 & 3 & 9 & 3 \end{bmatrix},\ c^2 = \begin{bmatrix} 5 & 4 & 4 & 7 \\ 6 & 1 & 2 & 4 \\ 3 & 6 & 3 & 1 \end{bmatrix},\ a = [\ 5\ 5\ 5\ ] \text{ and } b = [\ 4\ 3\ 4\ 4\ ].$$

Let $I = \{1, 2, 3\}$ be the set of sources and $J = \{4, 5, 6, 7\}$ the set of destinations. Suppose that the decision-maker wants to know an efficient solution in the region bounded by 57 in the value of the first objective function and 47 in the second one. So $\alpha_1 = 57$ and $\alpha_2 = 47$. Consider the weighting parameter value to be 0.1, so $c_{ij} = 0.1\ c^1_{ij} + 0.9\ c^2_{ij}\ (i \in I,\ j \in J)$.

Suppose that in a specific moment of the resolution we have the following basis:



corresponding to the basic solution

$$x = \begin{bmatrix} 1 & 0 & 4 & 0 \\ 2 & 3 & 0 & 0 \\ 1 & 0 & 0 & 4 \end{bmatrix}, \quad \begin{matrix} s_1 = 5 \\ s_2 = 4 \end{matrix}.$$

Let us assume that variables x_{26} and s_1 were selected to enter and leave the basis, respectively. The updated column corresponding to variable x_{26} is $y^2 = (5, -3)$. After step 6, the updated solution is

$$x = \begin{bmatrix} 2 & 0 & 3 & 0 \\ 1 & 3 & 1 & 0 \\ 1 & 0 & 0 & 4 \end{bmatrix}, \quad \begin{matrix} s_1 = 0 \\ s_2 = 7 \end{matrix}.$$

At this moment $Q^{-1} = I$ (I representing an 2×2 identity matrix). The next step, basis updating, illustrates case 7.(a).

7.(a) s_1 is an additional basic variable.

$$x^2 = \begin{bmatrix} x_{26} \\ s_2 \end{bmatrix} \text{ and } Q^{-1} = \begin{bmatrix} \frac{Q^{-1}_{11}}{y^2_1} & \frac{Q^{-1}_{12}}{y^2_1} \\ Q^{-1}_{21} - y^2_2 \frac{Q^{-1}_{11}}{y^2_1} & Q^{-1}_{22} - y^2_2 \frac{Q^{-1}_{12}}{y^2_1} \end{bmatrix} = \begin{bmatrix} \frac{1}{5} & 0 \\ \frac{3}{5} & 1 \end{bmatrix}$$

1. Determine an initial basic feasible solution

   Compute $Q_{2\times 2} : Q_{gh} = \overline{c}^{g}_{x^2_h}$, $g,h = 1,2$. Invert Q.

2. Determine the value for the dual variables

   - $\pi^2_h = \sigma_1 Q^{-1}_{1h} + \sigma_2 Q^{-1}_{2h}$, $h = 1,2$.
   - π^1_i , $i \in I$ and π^1_j , $j \in J$: $c_{ij} - c^1_{ij}\pi^2_1 - c^2_{ij}\pi^2_2 - \pi^1_i + \pi^1_j = 0$, $(i,j) \in B$

3. Determine the value of the reduced costs and test optimality

   - $\overline{c}_{ij} = c_{ij} - \pi^1_i + \pi^1_j - c^1_{ij}\pi^2_1 - c^2_{ij}\pi^2_2$, $i \in I, j \in J$
   - $\overline{c}_{s_r} = -\pi^2_r$, $r = 1,2$

   If $\overline{c}_{st} = \min\left(\overline{c}_{ij},\ i \in I, j \in J\ ;\ \overline{c}_{s_r},\ r = 1,2\right) \geq 0$, the current solution is optimal and the algorithm stops; otherwise, x_{st} is selected to enter the basis.

4. Update column corresponding to x_{st}

   $$y^1 :\ y^1_{ij} = \begin{cases} 1, & \text{if } (i,j) \text{ is in the direct path from } s \text{ to } t \text{ in } B \\ -1, & \text{if } (i,j) \text{ is in the inverse path from } s \text{ to } t \text{ in } B \\ 0, & \text{if } (i,j) \text{ is not in the path from } s \text{ to } t \text{ in } B \end{cases} \quad , \ (i,j) \in B$$

   $y^2 :\ y^2_h = Q^{-1}_{h1}\Delta_1 + Q^{-1}_{h2}\Delta_2$, $h = 1,2$

   For $h = 1,2$,

   $$y^1_{ij} \ \leftarrow \ y^1_{ij} + \begin{cases} -y^2_h, & \text{if } (i,j) \text{ is in the direct path from } h(i) \text{ to } h(j) \text{ in } B \\ y^2_h, & \text{if } (i,j) \text{ is in the inverse path from } h(i) \text{ to } h(j) \text{ in } B \\ 0, & \text{if } (i,j) \text{ is not in the path from } h(i) \text{ to } h(j) \text{ in } B \end{cases}$$

5. Determine the variable leaving the basis x_{ql}

   Let x_{ql} be the variable defining $\Delta = \min\left(\frac{x_{ij}}{y^1_{ij}} :\ y^1_{ij} > 0, i \in I, j \in J\ ;\ \frac{x^2_h}{y^2_h} :\ y^2_h > 0, h = 1,2 \right)$.

6. Update the current solution

   $$\begin{cases} x^1_{ij} \ \leftarrow \ x^1_{ij} - y^1_{ij}\Delta & , (i,j) \in B \\ x_{st} = \Delta & \\ x^2_h \ \leftarrow \ x^2_h - y^2_h\Delta & , h = 1,2 \end{cases}$$

7. Update the basis matrix $(\overline{B})$ of the current solution and Q^{-1}

   (a) If x_{ql} is an additional basic variable in position h

   $$\begin{array}{lllll} Q^{-1}_{h1} \ \leftarrow \ \frac{Q^{-1}_{h1}}{y^2_h} & , & Q^{-1}_{h2} \ \leftarrow \ \frac{Q^{-1}_{h2}}{y^2_h} & & \\ Q^{-1}_{g1} \ \leftarrow \ Q^{-1}_{g1} - y^2_g Q^{-1}_{h1} & , & Q^{-1}_{g2} \ \leftarrow \ Q^{-1}_{g2} - y^2_g Q^{-1}_{h2} & , & g \neq h \end{array}$$

   Return to 2.

   (b) Otherwise,

   i. If there is no cycle containing x_{ql} and x^2_h $(h = 1,2)$
   Update the tree and return to 2.

   ii. Otherwise,
   Let x^2_h be the additional basic variable, in position h, which forms a cycle with x_{ql}.
   Change x_{ql} with x^2_h. Update the tree.

   $$\mu_r = \begin{cases} 1, & \text{if } x_{ql} \text{ and the additional basic variable } r \text{ have the same direction in the cycle} \\ -1, & \text{if } x_{ql} \text{ and the additional basic variable } r \text{ have opposite directions in the cycle} \\ 0, & \text{if } x_{ql} \text{ and the additional basic variable } r \text{ do not form a cycle} \end{cases}$$

   $$\begin{array}{l} Q^{-1}_{h1} \ \leftarrow \ -\mu_1 Q^{-1}_{11} - \mu_2 Q^{-1}_{21} \\ Q^{-1}_{h2} \ \leftarrow \ -\mu_1 Q^{-1}_{12} - \mu_2 Q^{-1}_{22} \end{array}$$

   Return to 7.(a).

Figure 1: Scheme of the algorithm

2. Computing dual variables

$\delta_1 = 0 \quad \rightarrow \quad \delta_4 = \delta_1 - c_{14} = -5.1 \quad , \quad \delta_6 = \delta_1 - c_{16} = -3.8 \quad \rightarrow \quad \delta_2 = \delta_4 + c_{24} = 0.6$

x_2^2 is a slack variable

$\sigma_1 = c_{26} - \delta_2 + \delta_6 = 2.2 - 0.6 + (-3.8) = -2.2 \ \ ; \ \ \sigma_2 = c_{s_2} = 0$

$\pi_1^2 = \sigma_1 Q_{11}^{-1} + \sigma_2 Q_{21}^{-1} = -2.2 \times \frac{1}{5} + 0 \times \frac{3}{5} = -0.44 \ \ ; \ \pi_2^2 = \sigma_1 Q_{12}^{-1} + \sigma_2 Q_{22}^{-1} = -2.2 \times 0 + 0 \times 1 = 0$

$d_{ij} = c_{ij} - c_{ij}^1 \pi_1^2 \ , \ (i,j) \in B$

$d_{14} = 5.1 - 6(-0.44) = 7.74 \ , \ d_{16} = 3.8 - 2(-0.44) = 4.68 \ , \ d_{24} = 5.7 - 3(-0.44) = 7.02 \ ,$

$d_{34} = 3.2 - 5(-0.44) = 5.4 \ , \ d_{25} = 1.4 - 5(-0.44) = 3.6 \ , \ d_{37} = 1.2 - 3(-0.44) = 1.52$

$\pi_1^1 = 0 \quad \rightarrow \quad \pi_4^1 = \pi_1^1 - d_{14} = -7.74 \ , \ \pi_6^1 = \pi_1^1 - d_{16} = -4.68$

$\rightarrow \quad \pi_2^1 = \pi_4^1 + d_{24} = -0.72 \quad , \quad \pi_3^1 = \pi_4^1 + d_{34} = -2.34$

$\rightarrow \quad \pi_5^1 = \pi_2^1 - d_{25} = -4.32 \quad , \quad \pi_7^1 = \pi_3^1 - d_{37} = -3.86$

3. Computing reduced costs

$\bar{c}_{15} = c_{15} - \pi_1^1 + \pi_5^1 - c_{15}^1 \pi_1^2 = 3.7 - 0 + (-4.32) - 1 \times (-0.44) = -0.18 \ < \ 0$

$\bar{c}_{17} = c_{17} - \pi_1^1 + \pi_7^1 - c_{17}^1 \pi_1^2 = 6.8 - 0 + (-3.86) - 5 \times (-0.44) = 5.14$

$\bar{c}_{27} = c_{27} - \pi_2^1 + \pi_7^1 - c_{27}^1 \pi_1^2 = 3.7 - (-0.72) + (-3.86) - 1 \times (-0.44) = 1$

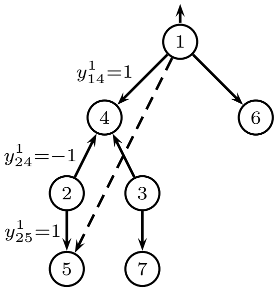
$\bar{c}_{35} = c_{35} - \pi_3^1 + \pi_5^1 - c_{35}^1 \pi_1^2 = 5.7 - (-2.34) + (-4.32) - 3 \times (-0.44) = 5.04$

$\bar{c}_{36} = c_{36} - \pi_3^1 + \pi_6^1 - c_{36}^1 \pi_1^2 = 3.6 - (-2.34) + (-4.68) - 9 \times (-0.44) = 5.22$

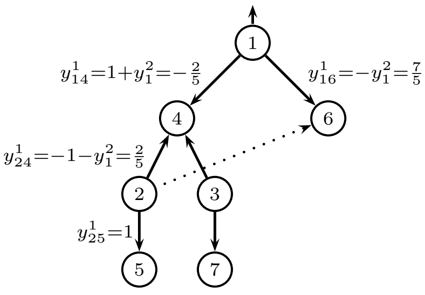
$\bar{c}_{s_1} = -\pi_1^2 = 0.44$

x_{15} is selected to enter the basis

4. Updating the column corresponding to the entering variable



$\Delta_1 = c_{15}^1 - y_{25}^1 c_{25}^1 - y_{24}^1 c_{24}^1 - y_{14}^1 c_{14}^1 = 1 - 5 + 3 - 6 = -7$
$\Delta_2 = c_{15}^2 - y_{25}^1 c_{25}^2 - y_{24}^1 c_{24}^2 - y_{14}^1 c_{14}^2 = 4 - 1 + 6 - 5 = 4$
$y_1^2 = Q_{11}^{-1} \Delta_1 + Q_{12}^{-1} \Delta_2 = \frac{1}{5} \times (-7) + 0 \times 4 = -\frac{7}{5}$
$y_2^2 = Q_{21}^{-1} \Delta_1 + Q_{22}^{-1} \Delta_2 = \frac{3}{5} \times (-7) + 1 \times 4 = -\frac{1}{5}$


1
y_{14}^1=1+y_1^2=−$\frac{2}{5}$
y_{16}^1=−y_1^2=$\frac{7}{5}$
4
6
y_{24}^1=−1−y_1^2=$\frac{2}{5}$
2
3
y_{25}^1=1
5
7

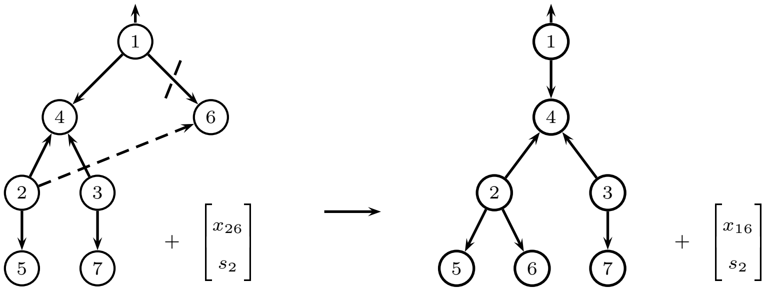
x_2^2 is a slack variable

5. Choosing the variable to leave the basis
$\Delta = \min\left(\frac{x_{24}}{y^1_{24}}, \frac{x_{16}}{y^1_{16}}, \frac{x_{25}}{y^1_{25}} \right) = \min\left(\frac{1}{2/5}, \frac{3}{7/5}, \frac{3}{1} \right) = \frac{15}{7} \Rightarrow x_{16}$ leaves the basis

6. Updating the solution

$$x = \begin{bmatrix} \frac{20}{7} & \frac{15}{7} & 0 & 0 \\ \frac{1}{7} & \frac{6}{7} & 4 & 0 \\ 1 & 0 & 0 & 4 \end{bmatrix} \quad s = \begin{bmatrix} 0 \\ \frac{52}{7} \end{bmatrix}$$

7.(b)*ii.* x_{16} belongs to the spanning tree and forms a cycle with x_{26}



$\mu_1 = -1\ ,\ \mu_2 = 0$
$Q^{-1}_{11} \leftarrow -\mu_1 Q^{-1}_{11} - \mu_2 Q^{-1}_{21} = -(-1)\frac{1}{5} - 0\ \frac{3}{5} = \frac{1}{5}$
$Q^{-1}_{12} \leftarrow -\mu_1 Q^{-1}_{12} - \mu_2 Q^{-1}_{22} = -(-1) \times 0 - 0 \times 1 = 0$
$x^2 = \begin{bmatrix} x_{15} \\ s_2 \end{bmatrix}$ and $Q^{-1} = \begin{bmatrix} \frac{Q^{-1}_{11}}{y^2_1} & \frac{Q^{-1}_{12}}{y^2_1} \\ Q^{-1}_{21} - y^2_2 \frac{Q^{-1}_{11}}{y^2_1} & Q^{-1}_{22} - y^2_2 \frac{Q^{-1}_{12}}{y^2_1} \end{bmatrix} = \begin{bmatrix} -\frac{1}{7} & 0 \\ \frac{4}{7} & 1 \end{bmatrix}$

Following algorithm steps 2 and 3 we can verify that all reduced costs are non negative so we are now in the optimal solution.

4 Computational Experience

Two versions of the algorithm described were implemented in *C++* language: one to support two additional restrictions (*Simplex++*) to be used in approach 1; and the other for one additional restriction only (*Simplex+*) to be used in approach 2.

We compared the performance of the algorithm developed with a general solver and introduce the algorithm in a decision support system. As a general solver we used ILOG Cplex 9.0 [10] with ILOG Concert 2.0 modelling language [11].

For the algorithms testing 85 instances were randomly generated with dimensions from 5 × 5 to 3500 × 3500, following a discrete uniform distribution. We generated 5 instances for each size with costs for the objective functions between 1 to 50, and all supply and demand values in the interval 1 to 20/40/65/80/100/200.

Some interactions with the decision-maker (DM) were simulated and the respective resulting problems were solved either by Cplex and the algorithm developed suitable to the interactive approach selected. Table 1 shows the mean value of the resolution times obtained. Column *Lex.Min.* in this

Algorithm	**Dimension**	**Lex.Min.**	**Int.Sol.**	**Interactions**			**Dimension**	**Lex.Min.**	**Int.Sol.**	**Interactions**		
Simplex++	5 × 5	0.0	0.0	0.0	0.0		500 × 1000	171.1	91.8	92.8	90.7	
Cplex		0.0	0.0	0.0	0.1			10.3	12.4	10.2	10.7	
Simplex+			0.0	0.0	0.0	0.0			118.0	96.5	77.8	60.8
Simplex++	10 × 10	0.0	0.0	0.0	0.0		1000 × 1000	510.3	278.4	293.0	279.3	
Cplex		0.0	0.0	0.0	0.0			49.7	37.6	37.3	49.8	
Simplex+			0.0	0.0	0.0	0.0			310.8	260.2	203.0	150.4
Simplex++	20 × 100	0.0	0.0	0.0	0.0		1000 × 2000	1 667.4	941.8	947.4	850.0	
Cplex		0.0	0.0	0.1	0.0			597.4	603.4	546.0	569.8	
Simplex+			0.0	0.0	0.0	0.0			912.4	731.4	561.0	412.0
Simplex++	50 × 100	0.0	0.0	0.0	0.0		1000 × 2000	1 709.4	972.0	980.4	877.1	
Cplex		0.1	0.0	0.0	0.0			577.0	537.4	508.6	527.7	
Simplex+			0.0	0.0	0.0	0.0			911.4	732.6	563.4	418.2
Simplex++	300 × 300	9.2	5.2	5.2	5.0		2000 × 2000	4 891.9	2 841.4	2 987.2	2 796.8	
Cplex		1.2	1.0	1.2	1.0			-	-	-	-	
Simplex+			8.2	7.2	6.0	5.0			2 321.0	1 797.8	1 343.0	986.6
Simplex++	300 × 500	22.1	11.2	12.0	11.8		2500 × 2500	10 652.6	6 492.2	6 532.0	5 787.3	
Cplex		2.3	2.0	2.2	2.3			-	-	-	-	
Simplex+			18.0	15.8	12.8	10.2			4 626.4	3 503.4	2 629.0	1 889.8
Simplex++	300 × 500	22.4	11.8	11.9	12.3		3000 × 3000	19 034.4	11 828.0	11 957.1	11 152.5	
Cplex		2.2	2.4	2.2	2.1			-	-	-	-	
Simplex+			17.8	15.0	12.8	9.8			7 461.6	5 598.8	4 092.6	2 961.4
Simplex++	500 × 500	58.4	35.8	47.6	37.5		3500 × 3500	33 060.0	21 261.0	20 132.3	18 912.8	
Cplex		4.7	3.8	3.8	4.6			-	-	-	-	
Simplex+			41.2	34.0	27.8	20.6			11 290.3	8 372.3	6 048.0	4 096.0
Simplex++	500 × 1000	167.0	89.4	94.4	90.6							
Cplex		10.8	12.8	10.8	10.3							
Simplex+			112.8	100.8	81.8	67.2						

Table 1: Resolution mean times (in seconds)

table corresponds to the calculation of the lexicographic minima (interactive algorithm initial step) determined either by Cplex and *Simplex++*. The lexicographic minima corresponding to the first objective optimal solution (z^{1*}) was used as an initial basic solution to *Simplex+*. Column *Int.Sol.* refers to an intermediate solution in the efficient region, computed using $\lambda = 0.5$ in the case of the first approach and $\alpha_1 = \frac{z_1^{2*} - z_1^{1*}}{2}$ in the case of the second approach (corresponds to half the efficient interval for the first objective function values). Column *Interactions* refer to the remaining simulated interactions. In the case of *Simplex++* and *Cplex* algorithms this last column is divided in two: the first one refers to the mean times obtained when the optimal solution is integer and the second one when the optimal solution is non-integer. In *Simplex+* algorithm this column is divided in three corresponding to each interaction simulated. For this approach we consider the DM always selects to optimize the second objective function restraining the first objective value to half the interval on the left side of the last upper bound considered. Thus, the upper bound value (α_1) is getting successively closer to the first objective value of the initial basic feasible solution used (z_1^{1*}).

We observed that for lower dimension instances, Cplex performs faster, but for larger dimension instances only *Simplex+* and *Simplex++* were able to solve the respective resulting problems. Cplex is not able to solve instances higher than 1000 × 2000.

When the optimal solution is non-integer, the algorithm developed has lower resolution times than for integer optimal solutions. For Cplex the opposite happens: the resolution times are higher when the optimal solution is non-integer.

Comparing the two approaches used, we concluded that the first one is faster on lower dimension instances, whereas the second approach is faster on higher dimension instances.

Finally, the second approach performs faster when the value of the initial basic solution is near the value of the upper bound on the additional constraint, α.

5 Decision Support System

In this section we show an overview of the Decision Support System developed. The purpose is to contribute with a practical tool to support the decision making in the presence of a problem with a transportation structure and two linear criteria with conflicting nature.

Based on Fernandes *et al.* work [8], this system was constructed following two main guidelines: a good quality interface, easily understandable and with an intuitive application; and a modular system allowing future integration of new models and algorithms. Moreover, it uses the basic functions of Microsoft Office *File*, such as *Save*, *Open*, *New*, *Print*, etc.

Generally, to initialize the system, the user must open a working area corresponding to its decision problem, either creating a new one (*New*) or opening a saved one (*Open*). This working area is divided into four folders: *Constants*, *Quantities*, *Costs* and *Results* (see figure 2).

The first three folders are devoted to the problem data introduction: in *Constants*, the number of sources and destinations (figure 2); in *Quantities*, the supply and demand values (figure 3); and finally in *Costs*, the cost coefficients of both objective functions (figure 4). The data can also be introduced through a data file outside the system, accessed by *Open* a working area. In this case the data file must be constructed following the order: number of sources and destinations; cost coefficients of the first objective function; supply values; cost coefficients of the second objective function; and demand values. All these values must be integer.

For academic purposes the data can also be randomly generated, pushing the *Random* button in the respective folder.

The interaction with the decision-maker (DM) is established in the *Results* folder (figure 5). The efficient solutions that are determined by the system can also be visualized here. This folder is composed by two windows: the left one represents the objective space, whereas the right one represents the decision space. Both may be visualized in a graphic or in a grid format, options performed by using the *Results* button. In order to facilitate the visualization in the graphic format of the objective space two buttons are available: the *Magnify* that has a magnifying glass effect; and the *Scale* which amplifies the object. The button *Original scale* returns to the original scale.

The *Execute* button enables the calculation of the efficient solutions. The first two times that this button is pushed the system shows the lexicographic minima of both objective functions corresponding to the initial step of the interactive algorithm from section 2 (figures 5 and 6). In case both solutions are equivalent the user is informed that there is no conflict between the two criteria and that the solution is optimal.

From this point on each time the *Execute* button is pressed a dialogue box emerges (figure 7) where the DM introduces the orientations for the efficient region search.

On the left side of this box the user has four distinct options. The first two correspond to approach 1 and the last two to approach 2. In *Upper bounds* the DM inserts a value for the upper bound of both criteria or in *Solution indexes* two solutions previously determined among which he/she intends to know a new non-dominated solution. In the next two options the user has the opportunity to know a solution with a specific value for one criteria (*First objective function (F1)=* and *Second objective function (F2)=*).

The right side (figure 7) is divided in two types of information. In (*Software*) and in order to solve the resulting problem of this dialogue phase, the user can choose between the algorithm developed in this work (*Network SIMPLEX*), the general solver Cplex (*General Solver*) or *MatLab 7.0.1 (R14).* If choosing the *Network SIMPLEX* the system will use either *Simplex++* or *Simplex+* depending on the option chosen on the left side. Note that *Cplex* is not supplied together with the system, thus this option will be available only in the presence of the respective user license. Finally, *Weights* (figure 7) allows the definition of the weighting parameter value when choosing the first approach on the left side. By default the weights are automatically defined (*Automatic*).

Solution 3 (figure 8) is a consequence of the options made in the dialogue box of figure 7. When a new non-dominated solution is introduced in the criteria space the system dashes two areas in grey. Those areas show regions where there are no more non-dominated solutions. On the upper right corner are the feasible solutions dominated by the introduced solution - this region is eliminated by dominance. The left down corner has no feasible solutions, otherwise it would dominate the introduced solution - this region is eliminated by non-feasibility.

Finally, all the determined efficient solutions may be accessed at any time through the pending listed box on the top of the objective space window. Therefore, it is possible for the user to revise the previously determined solutions.

6 Conclusion

In resume, with this work the authors contributed with an algorithm to solve transportation problems with additional linear constraints. Moreover the authors contributed with an interactive approach to bicriteria transportation problems and a decision support system that provides a practical tool to support the decision making.

Acknowledgements

The authors would like to thank Sérgio Fernandes for his contribution to the decision support system development.

References

[1] Aneja, Y.P., Nair, K.P.K. (1979). "Bicriteria transportation problem". *Management Science* 25, 73-78.

[2] Bazaraa, M.S., Jarvis, J.J., Sherali, H.D. (1990). "Linear programming and network flows". Wiley.

[3] Chen, S., Saigal, R. (1977). "A primal algorithm for solving a capacitated network flow with additional linear constraints". *Networks* 7, 59-79.

[4] Clímaco, J., Antunes, C.H., Alves, M.J. (1993). "Interactive decision support for multiobjective transportation problems". *European Journal of Operational Research* 65, 58-67.

[5] Diaz, J. (1978). "Solving multiobjective transportation problems". *Ekonomicko Mathematicky Obzor* 14, 267-274.

[6] Diaz, J. (1979). "Finding a complete description of all efficient solutions to a multiobjective transportation problem". *Ekonomicko Mathematicky Obzor* 15, 62-73.

[7] Ehrgott, M. (2005). "Multicriteria optimization". Springer.

[8] Fernandes, S., Captivo, M.E., Clímaco, J. (2005). "Um sistema de apoio à decisão para análise de problemas de localização bicritério". Working paper 2/05, Centro de Investigação Operacional, Faculdade de Ciências, Universidade de Lisboa. (in Portuguese)

[9] Gallagher, R.J., Saleh, O.A. (1994). "Constructing the set of efficient objective values in linear multiple objective transportation problems". *European Journal of Operational Research* 73, 150-163.

[10] ILOG (2003). "ILOG CPLEX 9.0 User's Manual". Incline Village, Nevada.

[11] ILOG (2004). "ILOG CONCERT 2.0 User's Manual". Incline Village, Nevada.

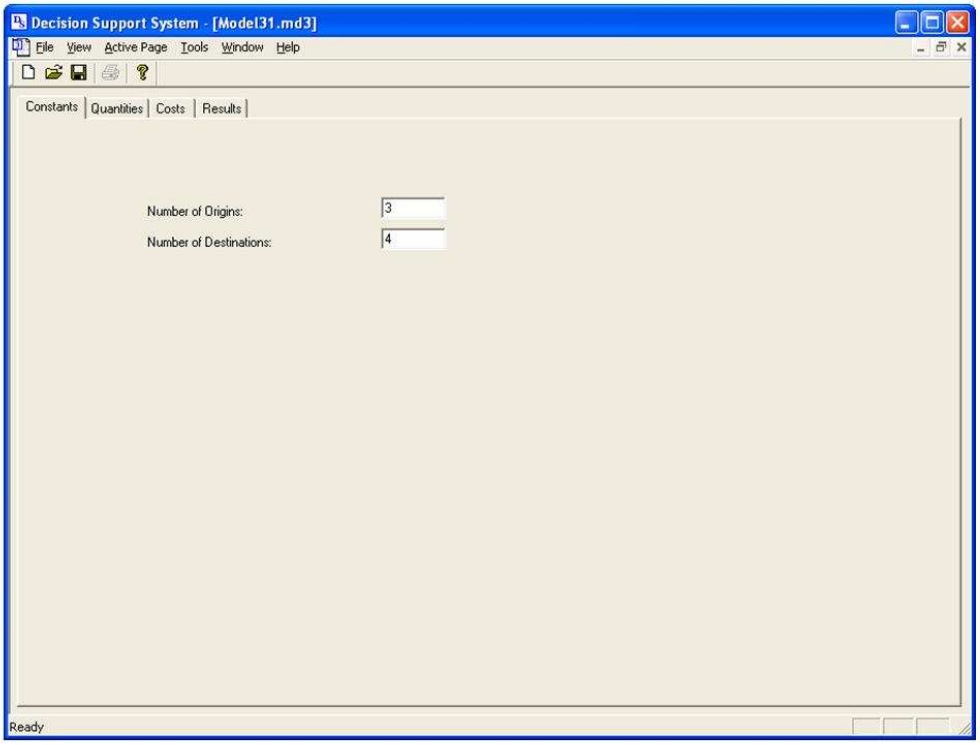
[12] Isermann, H. (1979). "The enumeration of all efficient solutions for a linear multipleobjective transportation problem". *Naval Research Logistics Quarterly* 26, 123-139.

[13] Lee, S.M., Moore, L.J. (1973). “Optimizing transportation problems with multiple objectives”. *AIIE Transactions* 5, 333-338.

[14] Ringuest, J.L., Rinks, D.B. (1987). “Interactive solutions for the linear multiobjective transportation problem”. *European Journal of Operational Research* 32, 96-106.

[15] Stewart, T.J., Ittmann, H.W. (1979). “Two-stage optimization in transportation problem”. *Journal of the Operational Research Society* 30, 897-904.

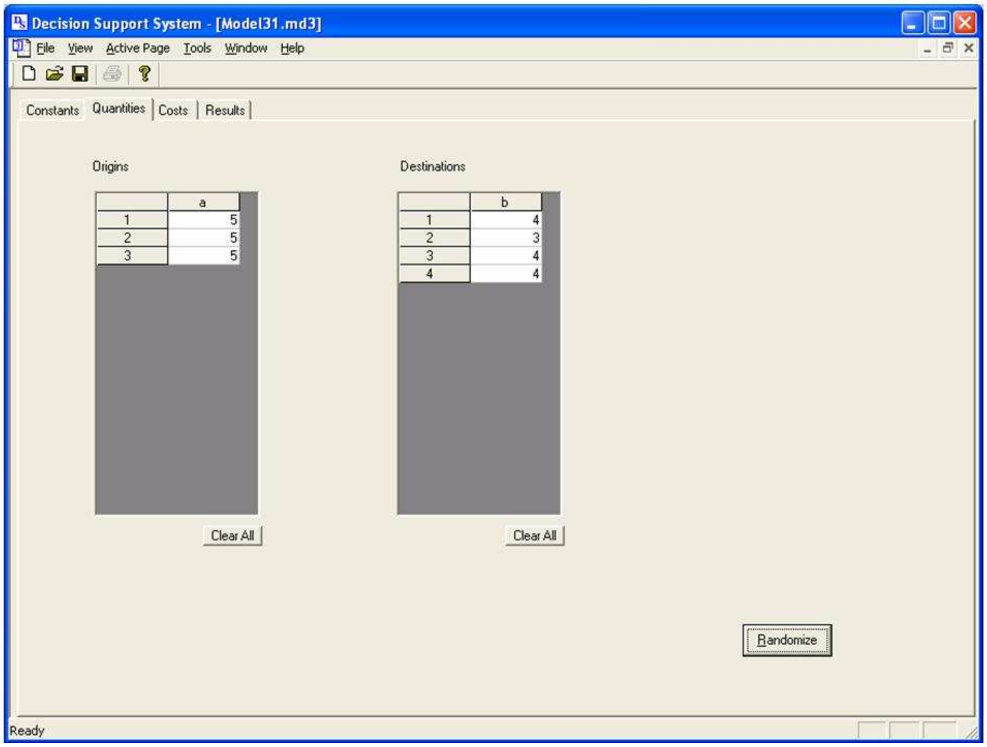
Appendix



Number of Origins: 3

Number of Destinations: 4

Figure 2: *Constants* folder.


Decision Support System - [Model31.md3]
File View Active Page Tools Window Help
Constants
Quantities
Costs
Results
Clear All
Clear All
Randomize
Ready

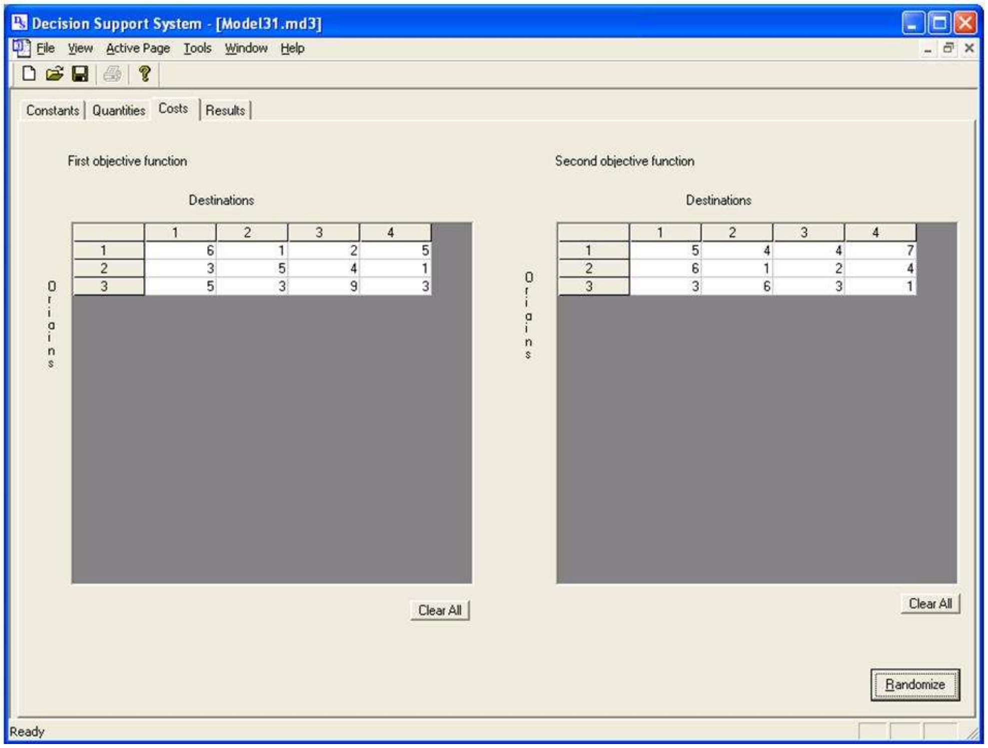
Origins

	a
1	5
2	5
3	5

Destinations

	b
1	4
2	3
3	4
4	4

Figure 3: *Quantities* folder.


Decision Support System - [Model31.md3]
File View Active Page Tools Window Help
Constants
Quantities
Costs
Results
Clear All
Clear All
Randomize
Ready

First objective function

Destinations

Origins

	1	2	3	4
1	6	1	2	5
2	3	5	4	1
3	5	3	9	3

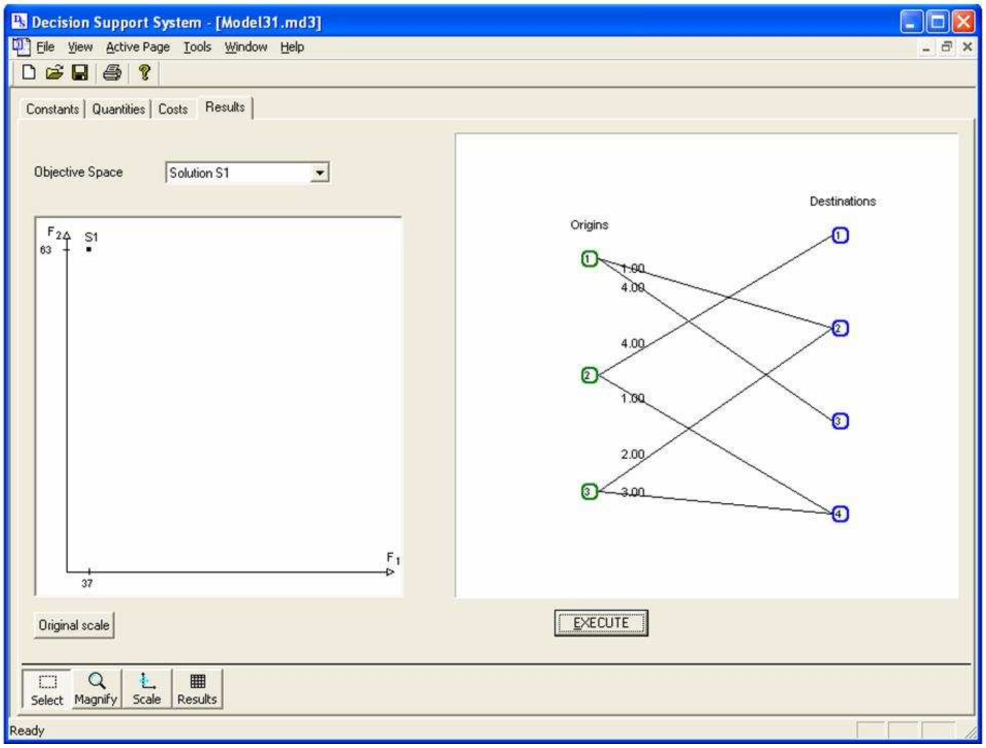
Second objective function

Destinations

Origins

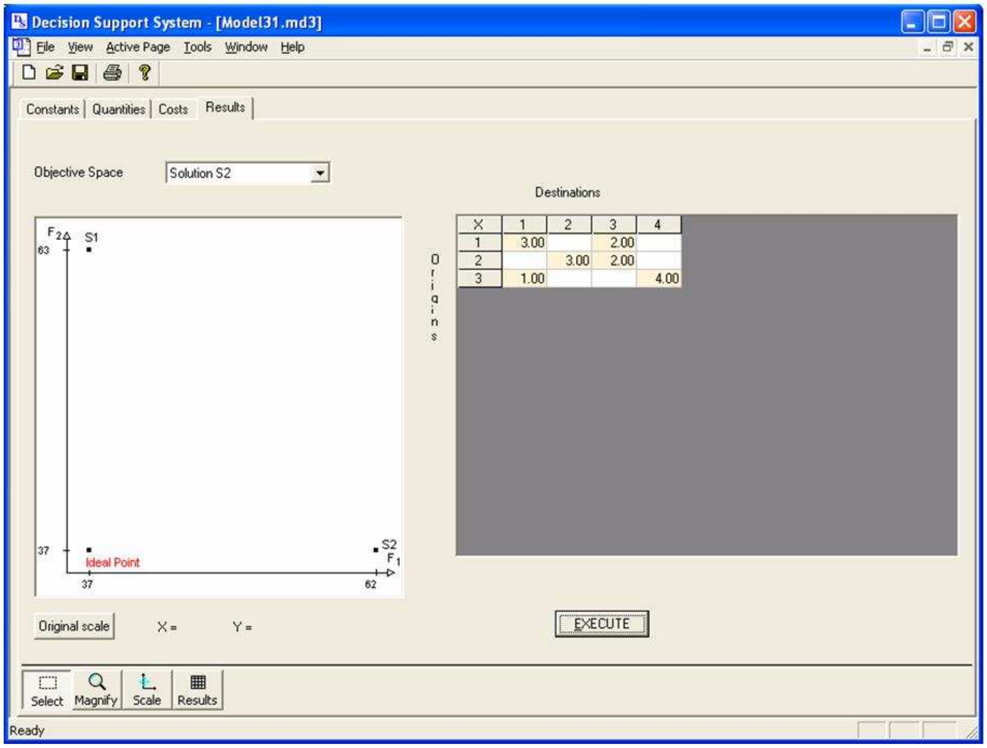
	1	2	3	4
1	5	4	4	7
2	6	1	2	4
3	3	6	3	1

Figure 4: *Costs* folder.


Decision Support System - [Model31.md3]
File View Active Page Tools Window Help
Constants Quantities Costs Results
Original scale
EXECUTE
Select Magnify Scale Results
Ready
F2
63
S1
37
F1
Origins
Destinations
1
2
3
4
1.00
4.00
4.00
1.00
2.00
3.00

Objective Space: Solution S1

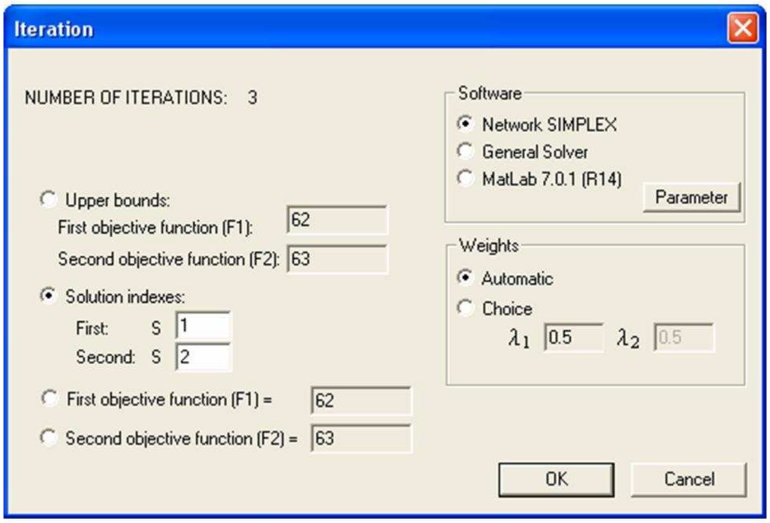
Figure 5: *Results* folder showing Solution 1 (first criteria lexicographic minima).


Decision Support System - [Model31.md3]
File View Active Page Tools Window Help
Constants Quantities Costs Results
Original scale
X =
Y =
EXECUTE
Select Magnify Scale Results
Ready
F2
63
S1
37
Ideal Point
S2
F1
37
62
Destinations
Origins

Objective Space: Solution S2

X	1	2	3	4
1	3.00		2.00	
2		3.00	2.00	
3	1.00			4.00

Figure 6: *Results* folder showing Solution 2 (second criteria lexicographic minima).



Iteration

NUMBER OF ITERATIONS: 3

○ Upper bounds:

First objective function (F1): 62

Second objective function (F2): 63

● Solution indexes:

First: S 1

Second: S 2

○ First objective function (F1) = 62

○ Second objective function (F2) = 63

Software

● Network SIMPLEX

○ General Solver

○ MatLab 7.0.1 (R14)

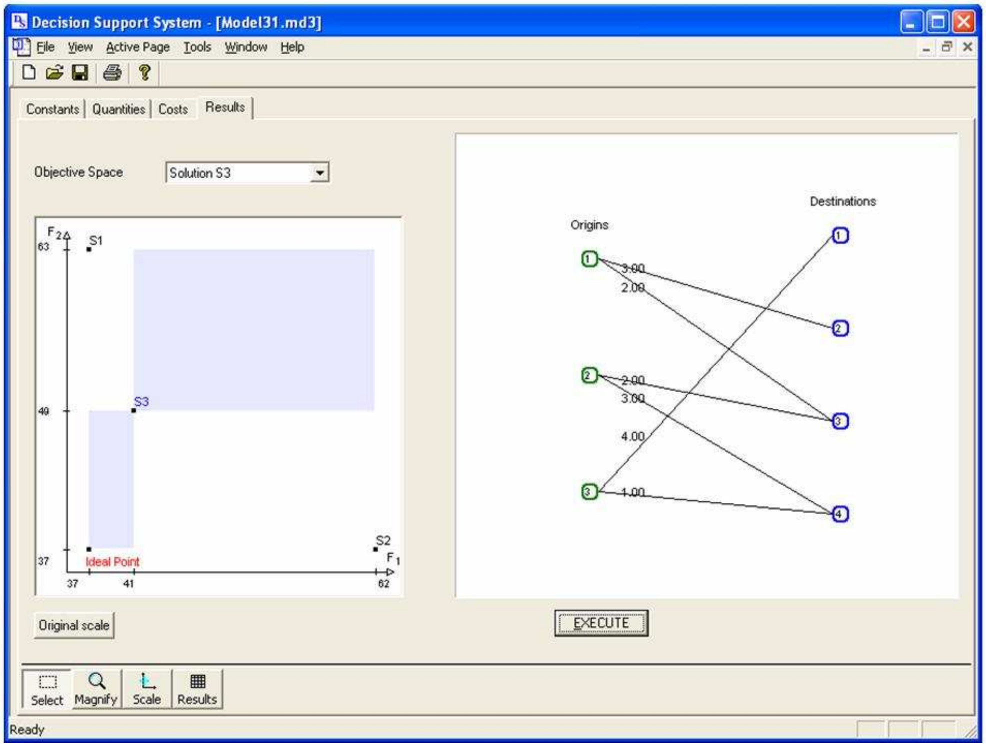
Weights

● Automatic

○ Choice

λ_1 0.5 λ_2 0.5

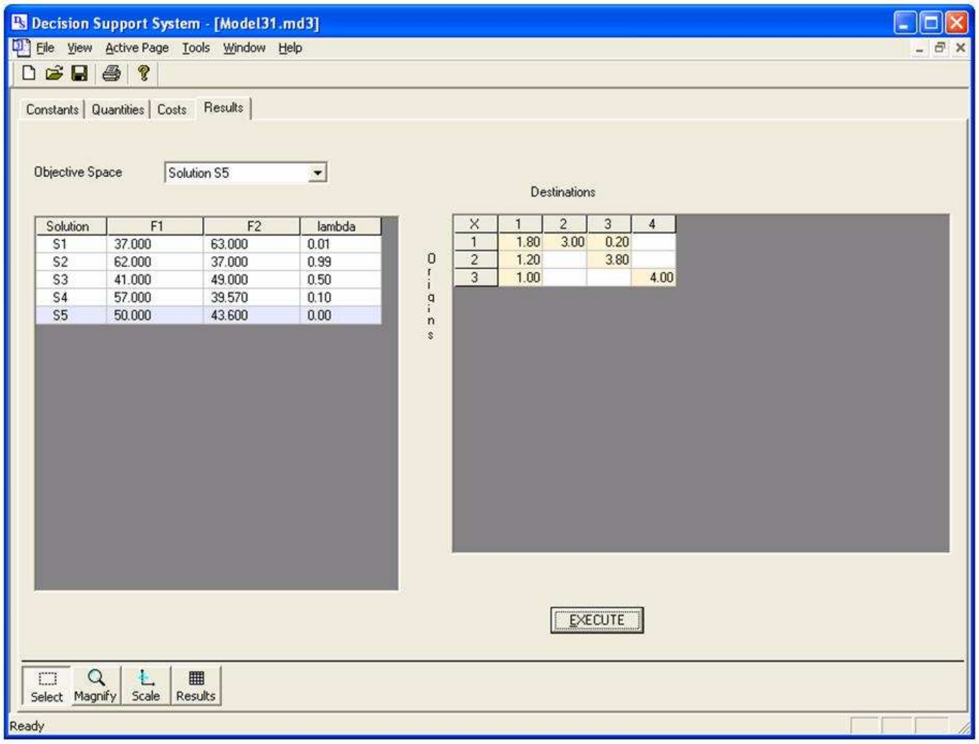
Figure 7: Dialog box for the decision-maker interaction.



Decision Support System - [Model31.md3]

Objective Space: Solution S3

Figure 8: *Results* folder showing Solution 3.


Decision Support System - [Model31.md3]
File View Active Page Tools Window Help
Constants Quantities Costs Results
EXECUTE
Select Magnify Scale Results
Ready

Objective Space: Solution S5

Solution	F1	F2	lambda
S1	37.000	63.000	0.01
S2	62.000	37.000	0.99
S3	41.000	49.000	0.50
S4	57.000	39.570	0.10
S5	50.000	43.600	0.00

Destinations

Origins

X	1	2	3	4
1	1.80	3.00	0.20	
2	1.20		3.80	
3	1.00			4.00

Figure 9: *Results* folder showing Solution 5.