Interval Multiobjective Solid Transportation Problem via Genetic Algorithms*

Fernando Jiménez
Dpto. Informática y Sistemas
Facultad de Informática
Universidad de Murcia
e-mail: fernan@dif.um.es

José L. Verdegay
Dpto. Ciencias de la Computación e I.A.
E.T.S.I. Informática
Universidad de Granada
e-mail: verdegay@robinson.ugr.es

Abstract

Transportation problem in which bounds are given on three item properties (source, destination and conveyance) is considered. Moreover, several objective functions are assumed, and the supplies, demands and conveyance capacities are, instead of point values, uncertain data expressed as intervals. One proposes a nonstandard genetic algorithm based solution method to solve the problem that extends other approaches considering only point values.

1 INTRODUCTION

Genetic Algorithms (GA) are adaptive procedures of optimization and search that find solutions to problems by an evolutionary process inspired in the mechanisms of natural selection and genetic. Currently, these algorithms are being highly considered above all in those problems with complex solution spaces for those which we do not have good algorithms to solve them. In many of these cases it would be desirable to have resolution methods that provide us with "good" solutions if they can be computed reasonably quickly. In a GA, a solution of our problem is called *individual*. In essence, it consists of maintaining a *population* of a given number of individuals, each one of them characterized by a "genetic" code (*genotype*) that identifies it univocally; thus an evolution of such population is simulated during the course of time, based on the apparition of new individuals resulting from *crossovers*, *mutations* and *direct reproductions* of the parents. An evaluation or objective function plays the role of the environment to distinguish in each generation that relatively "good" solutions reproduce, and that relatively "bad" solutions die, to be replaced by offspring of the good. Basically, we can say that a GA is based on the following components, for any type of application: a *"genetic" representation* of solutions to the problem; a way to create an initial population of solutions; an *evaluation function* to measure the fitness of any solution, and plays the role of the "environment", in which the better solutions (i.e., those that have a better objective value) may have greater probability of survival; *"genetic" operators* that effect the composition of children during reproduction; value for the *parameters* that the algorithm uses to guide its evolution: population size, number of generations, crossing and mutation probabilities, etc.

The classical GA use fixed-length binary strings and two basic genetic operators (binary mutation and binary crossover). However, it seems that a "natural" representation of a potential solution for a given problem plus a family of applicable "genetic" operators might be quite useful in the approximation of solutions of many problems. Moreover, the nontrivial constraints can be implemented incorporating problem-specific knowledge into the representation. Usually, this "modified" GA are called *Nonstandard Genetic Algorithms*, and more generally [12], *Evolution Programs*.

We can find extended documentation about Genetic Algorithms and its applications in [8][12].

2 INTERVAL MULTIOBJECTIVE SOLID TRANSPORTATION PROBLEM

The *Transportation Problem* (TP), originally developed by Hitchcock (1941), is one of the most common combinatorial problems involving constraints that has been studied. In the TP bounds are given on two item properties (source and destination). An extension of the clasical TP to the *Solid Transportation Problem* (STP) was stated by Shell [14] in 1955, in which three properties are considered in the constraint set. As third property usually is considered the type of product or the mode of transport (conveyance). Shell [14] also suggested the situations where the STP would arise, and four cases of STP are discussed according to the data given on the item properties. Haley [9] described the solution procedure of a STP, which basic-

*Research supported in part by DGICyT under project PB92-933.

ally consists in an extension of the modified distribution method for the classical TP. In this paper we have considered the three axial sums problem introduced by Shell, and mode of transport as third property.

In most of cases, it is required to solve the problem taking into account more than one decision criterion, thus giving place to the *Multiobjective Solid Transportation Problem* (MSTP). A variety of approaches has been developed by many authors for the Linear Multiobjective Transportation Problem ([1][2][7][11][13]) and recently, Bit et al [3] proposed a fuzzy programming approach to MSTP.

In all of the cases above cited, the supplies and demands (and conveyance capacities for the solid case) are given as point values, but often there is uncertainty on this input data, and decision makers are more comfortable expressing them as intervals. Chanas et al. [6] proposed a solution approach to the interval TP, and here we propose a NGA based solution method to the *Interval MSTP* (IMSTP) taking into account too the case in which the objective functions are nonlinear.

The posing of the IMSTP is as follows: A homogeneous product is to be transported from each of m sources to n destinations. The sources are production facilities, warehouses, or supply points, characterized by (uncertain) available capacities A_i, for $i = 1,\ldots,m$. The destinations are consumption facilities, warehouses, or demand points, characterized by (uncertain) required levels of demand B_j, for $j = 1,\ldots,n$. Let E_k $(k = 1,\ldots,l)$ be the (uncertain) amount of this product which can be carried by l different modes of transport or conveyances. A penalty $c^p_{ijk} \geq 0$ is associated with transportation of a unit of the product from source i to destination j by means of the k-th conveyance for the p-th decision criterion. The decision criteria could be transportation cost, delivery time, quantity of goods delivered, unfulfilled demand, underused capacity, reliability of delivery, safety of delivery (spoilage), and many others.

One must determine the amounts x_{ijk} of the product to be transported from all sources i to all destinations j by means of each conveyance k so that all P decision criteria are taken into account in a way satisfactory to the decision maker.

The IMSTP can be stated as:

$$
\begin{array}{ll}
Minimize & \\
Z_p = \displaystyle\sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{k=1}^{l} f^p_{ijk}(x_{ijk}), & for\ p = 1,\ldots,P \\
subject\ to: & \\
\displaystyle\sum_{j=1}^{n}\sum_{k=1}^{l} x_{ijk} =_1 A_i, & for\ i = 1,\ldots,m, \\
\displaystyle\sum_{i=1}^{m}\sum_{k=1}^{l} x_{ijk} =_1 B_j, & for\ j = 1,\ldots,n, \\
\displaystyle\sum_{i=1}^{m}\sum_{j=1}^{n} x_{ijk} =_1 E_k, & for\ k = 1,\ldots,l, \\
x_{ijk} \geq 0, & for\ all\ i,j,k,
\end{array}
\tag{1}
$$

where the subscript on Z_p and superscript on f^p_{ijk} denote the p-th decision criterion, $A_i = [a^1_i, a^2_i]$, $B_j = [b^1_j, b^2_j]$, $E_k = [e^1_k, e^2_k]$, for all i,j,k, are intervals of possible (admissible) values for the supplies, demands and conveyance capacities respectively, and the relation $=_1$ is formally defined as:

$$x =_1 [a,b] \underset{def}{\equiv} \exists z \in [a,b] : x = z$$

that is, $x =_1 [a,b]$ if and only if $x \in [a,b]$.

Thus, from the definition of $=_1$, problem (1) becomes:

$$
\begin{array}{ll}
Minimize & \\
Z_p = \displaystyle\sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{k=1}^{l} f^p_{ijk}(x_{ijk}), & for\ p = 1,\ldots,P \\
subject\ to: & \\
\displaystyle\sum_{j=1}^{n}\sum_{k=1}^{l} x_{ijk} \in A_i, & for\ i = 1,\ldots,m, \\
\displaystyle\sum_{i=1}^{m}\sum_{k=1}^{l} x_{ijk} \in B_j, & for\ j = 1,\ldots,n, \\
\displaystyle\sum_{i=1}^{m}\sum_{j=1}^{n} x_{ijk} \in E_k, & for\ k = 1,\ldots,l, \\
x_{ijk} \geq 0, & for\ all\ i,j,k.
\end{array}
\tag{2}
$$

The problem (2), and therefore (1) too, is feasible if and only if $D = [d^1, d^2] = A \cap B \cap E \neq \emptyset$, where A, B and E denote the following intervals:

$$A = \sum_{i=1}^{m} A_i = \left[\sum_{i=1}^{m} a^1_i, \sum_{i=1}^{m} a^2_i\right] = [a^1, a^2],$$

$$B = \sum_{j=1}^{n} B_j = \left[\sum_{j=1}^{n} b^1_j, \sum_{j=1}^{n} b^2_j\right] = [b^1, b^2],$$

$$E = \sum_{k=1}^{l} E_k = \left[\sum_{k=1}^{l} e^1_k, \sum_{k=1}^{l} e^2_k\right] = [e^1, e^2].$$

The STP is linear if the cost on a route is directly proportional to the amount transported ($f^p_{ijk}(x_{ijk}) = c^p_{ijk}x_{ijk}$, for all i,j,k,p); otherwise, it is nonlinear ($f^p_{ijk}(x_{ijk})$, for all i,j,k,p, are any type of practical or reasonable nonlinear function for the STP).

Note that the problem (2) generalize the classical TP with $P = 1$, $l = 1$, $f^p_{ijk}(x_{ijk}) = c^p_{ijk}x_{ijk}$ for all i,j,k,p, and $a^1_i = a^2_i$ for all i, $b^1_j = b^2_j$ for all j, $e^1_1 = e^2_1 = \sum_{i=1}^{m} a^1_i \left(= \sum_{i=1}^{m} a^2_i = \sum_{j=1}^{n} b^1_j = \sum_{j=1}^{n} b^2_j\right)$.

So that the problem is multiobjective, we should find an *optimal compromise solution*. An optimal compromise solution is a solution which is preferred by the decision maker to all other solutions, taking into consideration all criteria contained in the multiobjective functions. It is generally accepted that an optimal compromise solution has to be an *efficient* (*non-dominated*) solution. A feasible solution $\overline{X} = \{\overline{x}_{ijk}\}$ is said to be an efficient solution of the MSTP if there is no other feasible solution $X = \{x_{ijk}\}$ such that $\sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{k=1}^{l} f_{ijk}^{p}(x_{ijk}) \leq \sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{k=1}^{l} f_{ijk}^{p}(\overline{x}_{ijk})$, for $p = 1,\ldots,P$, and $\sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{k=1}^{l} f_{ijk}^{p}(x_{ijk}) < \sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{k=1}^{l} f_{ijk}^{p}(\overline{x}_{ijk})$, for at least one p. For a real word problem, the complete solution, i.e. set of all efficient solution, is not always necessary. We need only a method which finds an optimal compromise solution.

Although other multiobjective programming approaches can be used, we consider the *weighted additive model* [5][15]. From this approach, we suppose that decision maker can introduce a weight vector $\overline{w} = \{w_1,\ldots,w_P\}$, $0 \leq w_p \leq 1$, $p = 1,\ldots,P$, $\sum_{p=1}^{P} w_p = 1$, to reflect the relative importance of the objectives. Multiple objective functions are combined into one overall objective function Z, and the initial problem is then transformed as follows:

$$
\begin{aligned}
&Minimize\ Z = \sum_{p=1}^{P} w_p \sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{k=1}^{l} f_{ijk}^{p}(x_{ijk})\\
&subject\ to:\\
&\sum_{j=1}^{n}\sum_{k=1}^{l} x_{ijk} \in A_i, \quad for\ i = 1,\ldots,m,\\
&\sum_{i=1}^{m}\sum_{k=1}^{l} x_{ijk} \in B_j, \quad for\ j = 1,\ldots,n,\\
&\sum_{i=1}^{m}\sum_{j=1}^{n} x_{ijk} \in E_k, \quad for\ k = 1,\ldots,l,\\
&x_{ijk} \geq 0, \qquad for\ all\ i,j,k.
\end{aligned}
\tag{3}
$$

3 A NONSTANDARD GENETIC ALGORITHM FOR THE IMSTP

Michalewicz [12] proposed a NGA for the linear and nonlinear TP. For the STP, the most natural representation of a solution is a three-dimensional array of range $[m, n, l]$. Let us see the implications of this representation on constraint satisfaction, fitness function and genetic operators. First, we will describe a procedure to create an initial population of solutions:

```
procedure initial_population;
begin
  D = [d^1, d^2] := A ∩ B ∩ E;
  for s := 1 to pop_size do
  begin
    val =random real number ∈ [d^1, d^2];
    point_values(val,m,Ā,A,ā);
    point_values(val,n,B̄,B,b̄);
    point_values(val,l,Ē,E,ē);
    w := random boolean value;
    if w then initialization1(ā,b̄,ē,X)
    else initialization2(ā,b̄,ē,X);
    population[s]:=X;
  end;
end;
```

The point_values procedure obtain random admissible values of the supplies (demands and conveyance capacities) such that the total supply (total demand and total conveyances capacity) is equal to a random point value $val \in [d^1, d^2]$.

```
procedure point_values;
input: real value val; integer value s; interval array
X̄ = {X_1 = [x_1^1, x_1^2],...,X_s = [x_s^1, x_s^2]}; interval X =
[x^1, x^2];
output: real array x̄ = {x_1,...,x_s};
begin
  increm := val − x^1;
  for i := 1 to s do x_i := x_i^1;
  while increm > 0 do
  begin
    set all numbers from 1 to s as unvisited;
    repeat
      select an unvisited random number q from 1 to s;
      aux1 := min{(x_q^2 − x_q), increm};
      if all numbers are visited then
      begin
        x_q := x_q + aux1;
        increm := increm − aux1;
      end
      else
      begin
        aux2 :=random real number ∈ [0, aux1];
        x_q := x_q + aux2;
        increm := increm − aux2;
      end
    until all numbers are visited;
  end;
end;
```

We consider two ways to create a initial solution to the MSTP such that all constraints are satisfied. Initialization1 procedure will generate a solution that is at a vertex of the feasible polyhedron.

```
procedure initialization1;
input: real arrays ā = {a_1,...,a_m}, b̄ = {b_1,...,b_n},
ē = {e_1,...,e_l};
output: an array X = (x_ijk) (that represents a solu-
tion to the balanced point value MSTP);
```

```
begin
  set all numbers from 1 to m·n·l as unvisited;
  repeat
    select an unvisited random number q from 1 to
    m·n·1 and set it as visited;
    i := trunc((q − 1)/n·l + 1);
    j := trunc((q − 1) mod n·l/l + 1);
    k := (q − 1) mod n·l mod l + 1;
    val := min(a_i, b_j, e_k);
    x_ijk := val;
    a_i := a_i − val;
    b_j := b_j − val;
    e_k := e_k − val;
  until all number are visited;
end;
```

Initialization2 procedure obtain a solution that may be at an edge or a face of the feasible polyhedron.

procedure initialization2;
input: real arrays $\overline{a} = \{a_1, \ldots, a_m\}$, $\overline{b} = \{b_1, \ldots, b_n\}$, $\overline{e} = \{e_1, \ldots, e_l\}$;
output: an array $X = (x_{ijk})$ (that represents a solution to the balanced point value MSTP);

```
begin
  set all numbers from 1 to m·n·l as unvisited;
  repeat
    select an unvisited random number q from 1 to
    m·n·1 and set it as visited;
    i := trunc((q − 1)/n·l + 1);
    j := trunc((q − 1) mod n·l/l + 1);
    k := (q − 1) mod n·l mod l + 1;
    val_1 := min(a_i, b_j, e_k);
    If ((i is the last available row) or
    (j is the last available column) or
    (k is the last available "diagonal")) then
      val = val_1
    else
      val = random real number ∈ [0, val_1];
    x_ijk := val;
    a_i := a_i − val;
    b_j := b_j − val;
    e_k := e_k − val;
  until all number are visited;
  initialization1(a,b,e,debt);
  X:=X+debt; {sum of arrays}
end;
```

Constraint satisfaction: Any permutation of $m \cdot n \cdot l$ distinct numbers produces a unique solution that satisfies all constraints. This is guaranteed by initialization1 and initialization2 procudures. All genetic operators ensure feasible offsprings.

Fitness function: The fitness function is the objective function of the choosen auxiliary problem:

$$eval(x_{ijk}) = \sum_{p=1}^{P} w_p \sum_{i=1}^{m} \sum_{j=1}^{n} \sum_{k=1}^{l} f_{ijk}^{p}(x_{ijk})$$

Other evaluation functions are proposed in [4][10] according to the Fuzzy MSTP.

Genetic Operators: We use three genetic operators, mutation1, mutation2 and arithmetical crossover.

<u>Mutation1</u>: Let us denote a parent for mutation by $(m \times n \times l)$ array $V = (v_{ijk})$, and let us denote an offspring by $(m \times n \times l)$ array $V' = (v'_{ijk})$. We use the next algorithm to obtain the offspring:
Step 1. Set $V' = V$;
Step 2. Set randomly $I = \{I_1, \ldots, I_p\} \subset \{1, \ldots, m\}$ such that $1 \leq p \leq m$; $J = \{J_1, \ldots, J_q\} \subset \{1, \ldots, n\}$ such that $1 \leq q \leq n$; $K = \{K_1, \ldots, K_s\} \subset \{1, \ldots, l\}$ such that $1 \leq s \leq l$;
Step 3. Calculate point values a_{I_i}, b_{J_j} and e_{K_k}, $i = 1, \ldots, p$, $j = 1, \ldots, p$, $k = 1, \ldots, s$, for array V':

$$a_{I_i} = \sum_{j=1}^{n} \sum_{k=1}^{l} v'_{I_i jk}, \quad i = 1, \ldots, p,$$
$$b_{J_j} = \sum_{i=1}^{m} \sum_{k=1}^{l} v'_{i J_j k}, \quad j = 1, \ldots, q,$$
$$e_{K_k} = \sum_{i=1}^{m} \sum_{j=1}^{n} v'_{ij K_k}, \quad k = 1, \ldots, s;$$

Step 4. Create a $(p \times q \times s)$ subarray $W = (w_{ijk})$ from elements of the array V' in the following way:

$$w_{ijk} = v'_{I_i J_j K_k}, \; i = 1, \ldots, p, \; j = 1, \ldots, q, \; k = 1, \ldots, s;$$

Step 5. Calculate point values a_i^W, b_j^W and e_k^W, $i = 1, \ldots, p$, $j = 1, \ldots, q$, $k = 1, \ldots, s$, for array W:

$$a_i^W = \sum_{j=1}^{q} \sum_{k=1}^{s} w_{ijk}, \quad i = 1, \ldots, p,$$
$$b_j^W = \sum_{i=1}^{p} \sum_{k=1}^{s} w_{ijk}, \quad j = 1, \ldots, q,$$
$$e_k^W = \sum_{i=1}^{p} \sum_{j=1}^{q} w_{ijk}, \quad k = 1, \ldots, s;$$

Step 6. Calculate intervals $A_i^W = [a_i^{1W}, a_i^{2W}]$, $B_j^W = [b_j^{1W}, b_j^{2W}]$ and $E_k^W = [e_k^{1W}, e_k^{2W}]$, $i = 1, \ldots, p$, $j = 1, \ldots, q$, $k = 1, \ldots, s$, for array W as follows:

$$\left.\begin{array}{l} a_i^{1W} = max\{0, a_{I_i}^1 - a_{I_i} + a_i^W\} \\ a_i^{2W} = a_{I_i}^2 - a_{I_i} + a_i^W \end{array}\right\} i = 1, \ldots, p,$$

$$\left.\begin{array}{l} b_j^{1W} = max\{0, b_{J_j}^1 - b_{J_j} + b_j^W\} \\ b_j^{2W} = b_{J_j}^2 - b_{J_j} + b_j^W \end{array}\right\} j = 1, \ldots, q,$$

$$\left.\begin{array}{l} e_k^{1W} = max\{0, e_{K_k}^1 - e_{K_k} + e_k^W\} \\ e_k^{2W} = e_{K_k}^2 - e_{K_k} + e_k^W \end{array}\right\} k = 1, \ldots, s;$$

Step 7. Calculate $D^W = [d^{1W}, d^{2W}] = A^W \cap B^W \cap E^W$, where

$$A^W = \sum_{i=1}^{p} A_i^W = \left[\sum_{i=1}^{p} a_i^{1W}, \sum_{i=1}^{p} a_i^{2W} \right],$$

$$B^W = \sum_{j=1}^{q} B_j^W = \left[\sum_{j=1}^{q} b_j^{1W}, \sum_{j=1}^{q} b_j^{2W} \right],$$

$$E^W = \sum_{k=1}^{s} E_k^W = \left[\sum_{k=1}^{s} e_k^{1W}, \sum_{k=1}^{s} e_k^{2W} \right];$$

Step 8. Use the point_values procedure (with val =random real number $\in [d^{1W}, d^{2W}]$) to obtain new values a_i^W, b_j^W, e_k^W, $i = 1, \ldots, p$, $j = 1, \ldots, q$, $k = 1, \ldots, s$, for array W;
Step 9. Use the initialitation1 procedure to assign new values to the array W such that all constraints a_i^W, b_j^W and e_k^W are satisfied;
Step 10. Replace appropriate elements of array V' by new elements from the array W. In this way all global constraints A_i, B_j and E_k are preserved.

Mutation2: This version is identical to *Mutation1* except that in recalculating the contents of the choosen subarray W the initialization2 procedure is used.

Arithmetical crossover: We assume that two arrays $V_1 = (v_{ijk}^1)$ and $V_2 = (v_{ijk}^2)$ are selected as parents for the crossover operation. Then we produce two offspring $V_3 = (v_{ijk}^3)$ and $V_4 = (v_{ijk}^4)$ as follows:

$$V_3 = c_1 \cdot V_1 + c_2 \cdot V_2$$
$$V_4 = c_1 \cdot V_2 + c_2 \cdot V_1$$

where c_1 and c_2 are fixed ramdomly in each crossing operation, and $c_1, c_2 \geq 0$, $c_1 + c_2 = 1$. Since the constraints set is convex this operation ensures that both of children are feasible if both parents are.

4 NUMERICAL EXAMPLES

Example 1: Linear Interval Transportation Problem.

$m = 5$; $n = 4$.

Supplies:

$A_1 = [10, 15]$; $A_2 = [20, 30]$; $A_3 = [10, 35]$; $A_4 = [5, 10]$; $A_5 = [15, 20]$.

Demands:

$B_1 = [5, 10]$; $B_2 = [10, 30]$; $B_3 = [20, 40]$; $B_4 = [10, 20]$.

Total supply, total demand and intersection:

$A = [60, 110]$; $B = [45, 100]$; $A \cap B = [60, 100]$.

Costs:

i \ j				
	10	20	30	40
	50	20	30	60
	80	100	5	10
	15	10	30	10
	40	50	60	80

Solution:

a) Chanas et al [6]:

i \ j				
	10.0			
		20.0		
			15.0	5.0
				5.0
		10.0	5.0	

$$Z(x^*) = 1475.0$$

b) Genetic algorithm (population size = 20; generations number = 500; mutations number = 10; arithmetical crossovers number = 2):

i \ j				
	10.0			
		20.0		
			15.0	
				10.0
		10.0	5.0	

$$Z(x^*) = 1475.0$$

Example 2: Nonlinear Interval Solid Multiobjective Transportation Problem.

$m = 3$; $n = 3$; $l = 3$.

Supplies:

$A_1 = [5, 20]$; $A_2 = [10, 30]$; $A_3 = [5, 10]$.

Demands:

$B_1 = [15, 20]$; $B_2 = [10, 30]$; $B_3 = [20, 40]$.

Conveyance capacities:

$E_1 = [10, 15]$; $E_2 = [20, 30]$; $E_3 = [10, 30]$.

Total supply, total demand, total conveyances capacity and intersection:

$A = [20, 60]$; $B = [45, 90]$; $E = [40, 75]$; $A \cap B \cap E = [45, 60]$.

Objective functions:

$f_{ijk}^p(x_{ijk}) = c_{ijk}^p x_{ijk}^2$, for all i, j, k, p.

Costs:

$p = 1$

$i = 1$:

j \ k			
	30	20	15
	20	100	60
	75	50	45

$i = 2$:

j \ k			
	5	10	25
	60	45	30
	15	20	100

$i = 3$:

j \ k			
	40	15	35
	10	30	5
	5	100	50

$$p = 2 \left\{ \begin{array}{l}
\begin{array}{cc} & k \\ j & \left.\begin{array}{|c|c|c|}\hline 10 & 20 & 30 \\ \hline 25 & 40 & 60 \\ \hline 50 & 15 & 5 \\ \hline \end{array}\right\} i = 1 \end{array} \\[2ex]
\begin{array}{cc} & k \\ j & \left.\begin{array}{|c|c|c|}\hline 25 & 100 & 50 \\ \hline 10 & 10 & 20 \\ \hline 5 & 30 & 55 \\ \hline \end{array}\right\} i = 2 \end{array} \\[2ex]
\begin{array}{cc} & k \\ j & \left.\begin{array}{|c|c|c|}\hline 15 & 20 & 35 \\ \hline 20 & 45 & 60 \\ \hline 50 & 10 & 5 \\ \hline \end{array}\right\} i = 3 \end{array}
\end{array} \right.$$

Weights:

$w_1 = 0.5$; $w_2 = 0.5$.

Solution with genetic algorithms (population size = 20; generations number = 500; mutations number = 10; arithmetical crossovers number = 2):

$$\begin{array}{cc} & k \\ j & \left.\begin{array}{|c|c|c|}\hline 1.551 & 3.443 & 1.594 \\ \hline 1.265 & 0.950 & 0.581 \\ \hline 0.854 & 2.769 & 2.338 \\ \hline \end{array}\right\} i = 1 \end{array}$$

$$\begin{array}{cc} & k \\ j & \left.\begin{array}{|c|c|c|}\hline 2.289 & 1.326 & 1.051 \\ \hline 0.917 & 2.518 & 1.409 \\ \hline 5.613 & 3.726 & 0.806 \\ \hline \end{array}\right\} i = 2 \end{array}$$

$$\begin{array}{cc} & k \\ j & \left.\begin{array}{|c|c|c|}\hline 0.464 & 2.787 & 0.494 \\ \hline 0.683 & 1.220 & 0.456 \\ \hline 1.238 & 1.259 & 1.396 \\ \hline \end{array}\right\} i = 3 \end{array}$$

$$Z(x^*) = 2479.112$$

$$Z_1(x^*) = 2903.995$$

$$Z_2(x^*) = 2054.228$$

5 CONCLUSIONS

Other approach to solve the interval TP has been proposed by Chanas et al. [6] in which the problem is transformed into an auxiliary problem adding ficticius origins and destinations. The auxiliary problem is finally solved with a classical method for the TP. However, if this approach is applied to a nonlinear problem, is well known that classical method do not work so good. Moreover, this approach do not contemplate the solid case. In this paper we propose a genetic algorithm based alternative solution method for the IMSTP. The advantages of this method are: 1) there is not necessity of more additional memory than the one used for the TP with point values, 2) the solid case is contempled, and 3) it finds better solutions for the nonlinear problems.

References

[1] V.P. Aneja, K.P.K. Nair (1979). Bicriteria transportation problem. *Management Sci.* **25**: 73-78.

[2] A.K. Bit, M.P. Biswal, S.S. Alam (1992). Fuzzy programming approach to multicriteria decision making transportation problem. *Fuzzy Sets and Systems* **50**:135-141.

[3] A.K. Bit, M.P. Biswal, S.S. Alam (1993). Fuzzy programming approach to multiobjective solid transportation problem. *Fuzzy Sets and Systems* **57**:183-194.

[4] J.M. Cadenas, F. Jiménez (1994). A genetic algorithm for the multiobjective solid transportation problem: a fuzzy approach. *International Symposium on Automotive Technology and Automation, Procs. for the dedicated conferences on Mechatronics & Supercomputing Applications in the Transportation Industries* :327-334.

[5] V. Chankong, Y.Y. Haimes (1983). Multiobjective decision making: theory and methodology. North-Holland series in Systems Science and Engineering, Andrew P. Sage (ed.).

[6] S. Chanas, M. Delgado, J.L. Verdegay, M.A. Vila (1993). Interval and fuzzy extensions of classical transportation problems. *Transportation Planning and Technology* **17**(2):203-218. Special Issue: Application of Fuzzy Set Theory to Transportation.

[7] J.A. Diaz (1978). Solving multiobjective transportation problem. *Ekonomicko-Matematicky Obzor* **14**:267-274.

[8] D.E. Goldberg (1989). Genetic algorithms in search, optimization, and machine learning. Addison-Wesley.

[9] K.B. Haley (1963). The solid transportation problem. *Oper. Res.* **10**:448-463.

[10] F. Jiménez, J.M. Cadenas (1995). An evolutionary program for the multiobjective solid transportation problem with fuzzy goals. *Operations Research and Decision* **2**:5-20.

[11] S.M. Lee, L.J. Moore (1973). Optimizing transportation problem with multiple objectives. *AIIE Transactions* **5**:333-338.

[12] Z. Michalewicz (1992). Genetic algorithms+data structures=evolution programs. Springer Verlag.

[13] J.L. Ringuest, D.B. Rinks (1987). Interactive solutions for the Linear multiobjective transportation problem. *Eur. J. Oper. Res.* **32**:96-106.

[14] E. Shell (1955). Distribution of a product by several properties. *Directorate of Management Analysis, Procs. 2nd. Symposium in Linear Programming* **2**:615-642, DCS/Comptroller H.Q. U.S.A.F., Washington, D.C..

[15] M. Zeleny (1982). Multiple Criteria Decision Making. McGraw-Hill.