Military Technical College
Kobry Elkobbah,
Cairo, Egypt
May 25-27,2010

5th International Conference on
Mathematics and Engineering
Physics (ICMEP-5)

EM-18

Efficient Evolutionary Algorithm for solving Multiobjective Transportation Problem

A. A. Mousa[a,*] Hamdy M. Geneedy[a] Adel Y. Elmekawy[a]

Abstract

This paper presents an efficient evolutionary algorithm for solving multiobjective transportation problem MOTP. a new chromosome's structure was introduced, which is adopted as it is capable to representing all possible feasible solutions. Also, in order to keep the feasibility of the chromosome, a criterion of the feasibility was designed. Based on this criterion the crossover and mutation were implemented and they can always generate feasible chromosomes. To avoid an overwhelming number of solutions the algorithm maintains a finite-sized archive of non-dominated solutions, which gets iteratively updated in the presence of new solutions based on the concept of Epsilon-dominance. Epsilon dominance process saves the most representative solutions. Finally, we report numerical results in order to establish the actual computational burden of the proposed algorithm and to assess its performances with respect to classical approaches for solving MOTP**.**

1. Introduction

The transportation problem, a specific problem of resource allocation, can be formulated as a linear programming problem, where the constraints have a special structure. In its classical form the transportation problem minimizes the cost of transporting some commodity that is available at m sources (supply nodes) and required at n destinations (demand nodes). The source parameter (a_i) may be production facilities, warehouse, etc., whereas the destination parameter (b_i) may be warehouse, sales outlet, etc. The penalty (C_{ij}) that is, the co-efficient of the objective functions, could represent transportation cost, delivery time, number of goods transposed, unfulfilled demand, and many others. Thus multiple penalty criteria may exist concurrently which leads to the research work on multiobjective transportation problem.

Until now, many researchers also have a great interest in the multiobjective transportation problem, and a number of methods had been proposed for solving it [1-4,9,14,19-20].

[*] Corresponding author
Dr Abd allah Mousa (E-mail : A_mousa15@yahoo.com)

A large class of interesting problems, including many optimization problems, has no reasonably fast, guaranteed algorithms for solution. In some applications a near optimal solution is acceptable if it can be computed reasonably quickly; one approach to finding such solutions is to use a Population based algorithms that, given sufficient time, can find solutions as close to the real optimum as we wish. Michalewicz et al. [11,19] firstly discussed the use of genetic algorithm (GA) for solving linear and nonlinear transportation problems. They used these problems as an example of constrained optimization problems, and investigated how to handle such constraints with GA. The matrix representation was used to construct a chromosome and designed the matrix-based crossover and mutation in their investigation. Gen et al. [6] further extended Michalewicz's work to bicriteria solid transportation problem. They embedded the basic idea of criteria space approach in evaluation phase so as to force genetic algorithm towards exploiting the nondominated points in the criteria space.
Also, Gen et al. [7] have proposed a new approach which is spanning tree-based genetic algorithm for solving MOTP. Spanning tree-based encoding was implemented with Prüfer number and adopted to represent a balanced transportation solution.

However, in more real world transportation problems to be solved have a large scale. for the problem with *m* sources and *n* destinations, for a solution, the matrix-based representation require $m \times n$ memories in the evolution process. it requires so much memories in the development of implementation, and will be spend more computational time. Also, some of multiobjective evolutionary algorithm suffer from the large size problem of the Pareto set e.g.[14]. Therefore some methods have been proposed to reduce the Pareto set to a manageable size. However, the goal is not only to prune a given set, but rather to generate a representative subset, which maintains the characteristics of the generated set.

In this paper we present an improved evolutionary algorithm to solve MOTP, where chromosome's structure was designed so that the feasibility can be preserved. In order to keep the feasibility of the chromosome, a criterion of the feasibility was designed. Based on this criterion the crossover and mutation were implemented and they can always generate feasible chromosome. The algorithm is an iterative multiobjective algorithm with an external population of Pareto optimal solutions that best conform a Pareto Front.
Finally, we report numerical results in order to establish the actual computational burden of the proposed algorithm and to assess its performances with respect to classical algorithms for solving MOTP.

This paper is organized as follows; multiobjective optimization are reviewed in section 2. Section 3 gives out the definition of MOTP. The original algorithm is presented in section 4. Experimental, results and discussions are discussed in section 5. Conclusion follows in section 6.

2. MULTIOBJECTIVE OPTIMIZATION

A general multiobjective optimization problem is expressed by:
MOP :

$$\begin{aligned} \text{Min} \quad F(x) &= (\ f_1(x), f_2(x), \ldots, f_k(x))^T \\ \text{s.t.} \quad & x \in S \\ x &= (x_1, x_2, \ldots, x_n)^T \end{aligned}$$

Where $(f_1\ (x), \ldots, f_k\ (x))$ are the *k* objectives functions, $(x_1, x_2, \ldots, x_n)$ are the n optimization parameters, and $S \in R^n$ is the solution or parameter space.

Definition 1.[13] (Pareto optimal solution): x^* is said to be a Pareto optimal solution of MOP if there exists no other feasible x (i.e., $x \in S$) such that, $f_j(x) \le f_j(x^*)$ for all $j = 1, 2, ..., k$ and $f_j(x) < f_j(x^*)$ for at least one objective function f_j .

Definition 2 [8]. (ε-dominance) Let $f : x \to R^k$ and $a,b \in X$. Then a is said to ε-dominate b for some ε > 0, denoted as $a \succ_\varepsilon b$, if and only if for $i \in \{1,...,k\}$
$(1+\varepsilon) f_i(a) \ge f_i(b)$

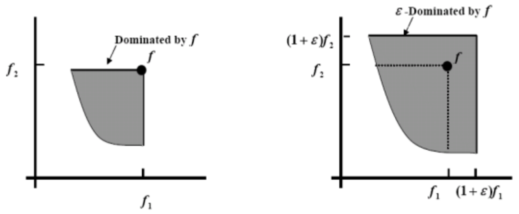


Fig. 1. Graphs visualizing the concepts of dominance (left) and ε-dominance (right).

Definition 3 [8]. (ε-approximate Pareto set) Let x be a set of decision alternatives and $\varepsilon > 0$. Then a set x_ε is called an ε-approximate Pareto set of X , if any vector $a \in x$ is ε-dominated by at least one vector $b \in x_\varepsilon$, i.e.,
$\forall a \in x : \exists b \in x_\varepsilon$ such that b $\succ_\varepsilon a$

According to definition 2, the ε value stands for a relative "tolerance" allowed for the objective values (Fig.1). This is especially important in higher dimensional objective spaces, where the concept of ε-dominance can reduce the required number of solutions considerably. Also, the use of ε -dominance also makes the algorithms practical by allowing a decision maker to control the resolution of the Pareto set approximation by choosing an appropriate ε value

3. Multiobjective Transportation Problem (MOTP)

In real-life situations, the transportation problem (TP) usually involves multiple, conflicting, and incommensurate objective functions. This type of problem is called multiobjective transportation problem (MOTP). The mathematical model of MOTP can be stated as follows:

$$F^k(x) = \sum_{i=1}^{m} \sum_{j=1}^{n} C_{ij}^k x_{ij},$$

Subject to

$$\sum_{j=1}^{n} x_{ij} = a_i, i = 1, 2, .., n$$

$$\sum_{i=1}^{m} x_{ij} = b_i, j = 1, 2, .., m$$

$$x_{ij} \ge 0 \quad \forall i, j$$

where $F^k(x) = \{F^1(x), F^2(x), ..., F^k(x)\}$ is a vector of *K* objective functions, the superscript on both $F^k(x)$ and C_{ij}^k are used to identify the number of objective functions (k = 1, 2, . . . , K), and m and n are the number of sources and destinations, respectively.

The above problem implies that the total supply $\sum_{i=1}^{m} a_i$ must be equal to the total demand $\sum_{j=1}^{n} b_j$. when total supply is equals to the total demand (i.e., total flow) the resulting formulation is called a balanced transportation problem. In this paper, we assume a balanced transportation problem, where the unbalanced transportation problem can be converted to a balanced transportation problem after including a dummy origin or a dummy destination. The solution of this problem is called a nondominated solution (if we refer to the objective function) and an efficient solution (if we refer to the decision variables space).

4. The Proposed Algorithm

Genetic algorithms [5,6,11,15] are such a class of evolutionary based algorithms that start with a population of randomly generated candidates and "evolve" toward better solutions by applying genetic operators, modeled on the genetic processes occurring in nature. It is generally accepted that any multiobjective genetic algorithm to solve a problem must have five components:

1. A genetic representation of solutions,
2. A way to create an initial population,
3. An evaluation function that plays the role of the environment,
4. Genetic operators that effect the composition of children during reproduction,
5. Values for the parameters that the genetic algorithm uses (population size, probabilities of applying genetic operators, etc.)
6. A way to update the archived Pareto solutions.

In the following sub-sections, we present an improved evolutionary algorithm for solving the MOTP.

4.1. Initialization Stage

The genetic representation is a kind of data structure which represents the candidate solution of the problem in coding space. In order to form the appropriate design of chromosome, first consider each chromosome consists of a sequence of m sub-chromosome (*m* is the number of supplies). Each sub-chromosome (Fig.2) consists of *n* genes (n is the number of demands). all chromosome are generated randomly such that the sum of total genes of each sub-chromosome equal to the corresponding supply amount. That is $\sum_{j=1}^{n} Gene_{ij} = a_i$ for each sub-chromosome *i*.

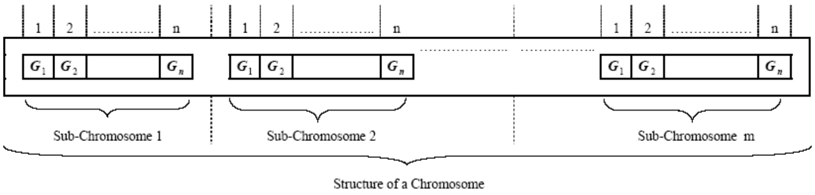
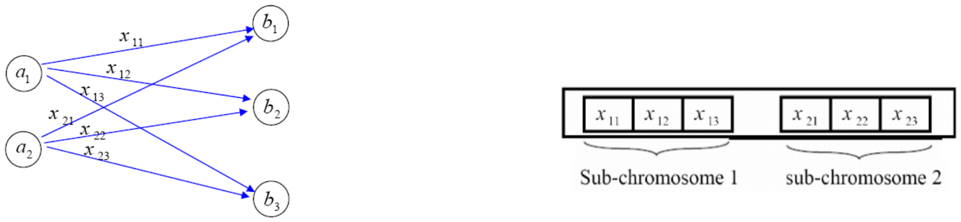


Fig. 2. Structure of chromosome for MTP with *n* sources and *m* destinations

In the example problem in figure 3, we have two supplies (*m=2*) and three demands (*n=3*). In order to design the appropriate structure of chromosome using the proposed algorithm, first consider each chromosome consists of two sub chromosome (*m=2*), Each sub-chromosome consists of *3* genes (*n=3*). we generate all chromosome randomly such that, $x_{11} + x_{12} + x_{13} = a_1$, $x_{21} + x_{22} + x_{23} = a_2$.



(a) Transportation graph

(b) Chromosome structure

Fig. 3. Illustration of chromosome's representation.

It is interesting here to note that all generated chromosome has the following characteristics.
1. All of generated individual are feasible.

2. The chromosome length is only $m \times n$, that is $\sum_{i=1}^{m}\sum_{j=1}^{n} Gene_{ij} = \sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j$, where the problem has *m* sources and *n* destinations.

4.2. Evaluation of Non-Dominated Solutions

A population of size N can be evaluated according to non-domination concept. Consider a set of population members, having *K (K>1)* objective function values. the following procedure explains the algorithm by which the nondominated set of solutions can be found[18].
Step 0: Begin with i=1.
Step 1: For all $j = 1, 2, .., N$ and $j \neq i$, compare solutions x^i and x^j for domination.
Step 2: If for any j . x^i, is dominated by x^j, mark x^i as "dominated".
Step 3: If all solutions (that is, when $i = N$ is reached) in the set are considered, Go to Step 4, else increment i by one and Go to Step 1.
Step 4: All solutions that are not marked "dominated" are non-dominated solutions.

The algorithm initially locates an externally finite size archive of observed nondominated solutions.

4.3. Selection Stage

Selection (reproduction) operator is intended to improve the average quality of the population by giving the high-quality chromosomes a better chance to get copied into the next generation. The selection directs GA search towards promising regions in the search space. We propose a random-weight approach [16] to obtaining a variable search direction towards the Pareto frontier. Suppose that we are going to maximize *k* objective function. The weighted-sum objective is given as follows:

$$f(x) = w_1 f_1(x) + ... + w_k f_k(x) = \sum_{i=1}^{k} w_i f_i(x)$$

where x is a string (i.e., individual), $f(x)$ is a combined fitness function, $f_i(x)$ is the *i-th* objective function and $\left\{ w_i \mid \sum_{i=1}^{k} w_i = 1 \right\}$ is a constant weight for $f_i(x)$.
We employ roulette wheel selection as selection mechanism in this study. Where, the individuals on each generation are selected for survival into the next generation according to a probability value proportional to the ratio of individual fitness over total population fitness; this means that on average the next generation will receive copies of an individual in

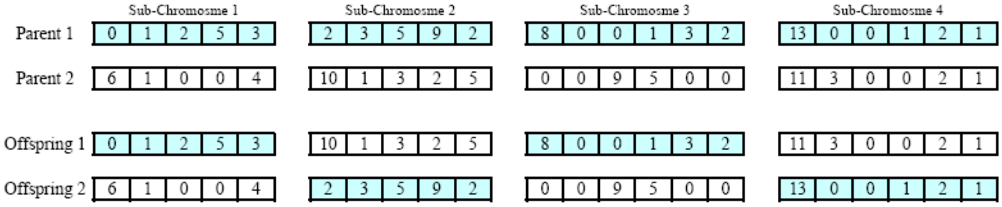
proportion to the importance of its fitness value. The probability of variable selection is proportional to its fitness value in the population, according to the formula given by

$$p(x)=\frac{f(x)-f_{Min}(\psi)}{\sum_{x\in\psi}\{f(x)-f_{Min}(\psi)\}},$$

where $p(x)$, selection probability of a string x in a population ψ and $f_{Min}(\psi)=Min\{f(x)\,|\,x\in\psi\}$

4.4. Crossover Operators

The goal of crossover is to exchange information between two parents chromosomes in order to produce two new offspring for the next population, we present a modified uniform crossover, where one offspring is constructed by choosing every sub-chromosome with a probability P (usually $P=0.5$ is used) from either parent, as shown in figure 4.
In the example problem in figure 4, we have four supplies (that is, we have four sub-chromosomes). The second and fourth sub-chromosome are exchanged between parents. It is interesting here to note that all offspring's chromosome are feasible.

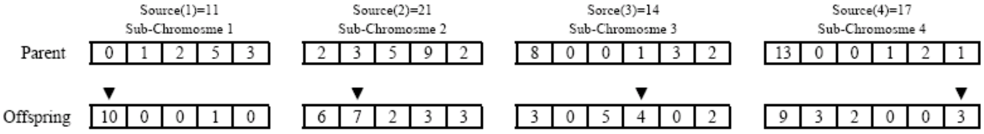


	Sub-Chromosme 1	Sub-Chromosme 2	Sub-Chromosme 3	Sub-Chromosme 4
Parent 1	0 1 2 5 3	2 3 5 9 2	8 0 0 1 3 2	13 0 0 1 2 1
Parent 2	6 1 0 0 4	10 1 3 2 5	0 0 9 5 0 0	11 3 0 0 2 1
Offspring 1	0 1 2 5 3	10 1 3 2 5	8 0 0 1 3 2	11 3 0 0 2 1
Offspring 2	6 1 0 0 4	2 3 5 9 2	0 0 9 5 0 0	13 0 0 1 2 1

Fig. 4. Graphs visualizing the crossover operators.

4.5. Mutation Operators

A mutation operator is a random process where one genotype is replaced by another to generate a new chromosome. Such a mutation operator first select a gene randomly from *ith* sub-chromosome and then replace it with a random integer within the interval of $[0,a_i]$, all other genes in *ith* sub-chromosome are generated such that the sum of all genes in the *ith* sub-chromosome equal to the *ith* supply $\sum_{j=1}^{n}Gene_{ij}=a_i$, as shown in figure 5.



	Source(1)=11 Sub-Chromosme 1	Source(2)=21 Sub-Chromosme 2	Sorce(3)=14 Sub-Chromosme 3	Source(4)=17 Sub-Chromosme 4
Parent	0 1 2 5 3	2 3 5 9 2	8 0 0 1 3 2	13 0 0 1 2 1
Offspring	10 0 0 1 0	6 7 2 3 3	3 0 5 4 0 2	9 3 2 0 0 3

Fig. 5: Graphs visualizing the mutation operators.

In the example problem in figure 5, we have four supplies, $a_1=11$, $a_2=21$, $a_3=14$ and $a_4=17$ the first gene in the first sub-chromosome is mutated (random integer $[0,a_1]=[0,11]$) and all other genes in the 1st sub chromosome are generated such that the sum of all genes equal to the supply amount $\sum_{j=1}^{5}G_j=a_1\Rightarrow\sum G_1+G_2+G_3+G_4+G_5=11$. Through this mutation operator, the population's feasibility was preserved.

4.6. Update Function for Epsilon-Pareto Set.

Algorithm 1(Table I) shows the structure of the proposed algorithm. The purpose of the function generate is to generate a new population in each iteration t, using the contents of the old population $P^{(t-1)}$ and the old archive set $A^{(t-1)}$ in association with the result of recombination and mutation of parents. The function update gets the new population $P^{(t)}$ and the old archive set $A^{(t-1)}$ and determines the updated one, namely $A^{(t)}$ .This is explained in algorithm 2.

Algorithm 2 (Table II) is a two level concept. On the coarse level, the search space is characterized by division boxes [15], where each vector belongs to one box. On the fine level at most one element is kept in each box.

As a result the proposed algorithm which is based on the (GA) uses a finite memory, successively updates a finite subset of vectors that ε -dominate all vectors generated so far. It guarantees that the subset contains only one element which are not dominated by any of the generated vectors. This puts limits to the size of the archive according the selected value of ε . Accordingly the algorithm is more practical where a decision maker is able to control the resolution of the Pareto set approximation according his needs. Also it guarantees an optimal distribution of solutions [15]. The algorithm has a low computational time where, the computational time grows with the number of archived solutions. The proposed algorithm enable to consider many objective functions. Accordingly it provide the facility to consider more criteria in RPC problem such as maximum voltage deviation .

TABLE 1 ALGORITHM 1

```
$t \leftarrow 0$
Create $P^{(0)}, R^{(0)}$
$A^{(0)} = \text{nondominated}(P^{(0)})$
while terminate $(A^{(t)}, t) = false$ do
    $t \leftarrow t + 1$
    $P^{(t)} \leftarrow \text{generate}(A^{(t-1)}, P^{(t-1)})$ {generate new search point}
    $A^{(t)} \leftarrow \text{update}(A^{(t-1)}, P^{(t)})$ {update archive (algorithm 2)}
end while
Output : $A^{(t)}$
```

TABLE 2 ALGORITHM 2

```
INPUT A,x
$D \leftarrow \{x' \in A : box(x) \succ box(x')\}$
if $D \neq \phi$ then
    $A' \leftarrow A \cup \{x\} \setminus D$
else if $\exists x' : (box(x') = box(x) \wedge x \succ x')$ then
    $A' \leftarrow A \cup \{x\} \setminus \{x'\}$
else if $\nexists x' : (box(x') \succeq box(x))$ then
    $A' \leftarrow A \cup \{x\}$
else
    $A' \leftarrow A$
endif
OUTPUT $A'$
```

5. Experimental, Results and Discussions

The proposed algorithm was implemented on 2.7-MHz PC using MATLAB 6.5. To confirm the effectiveness of the algorithm on the transportation problem, three numerical problems were used in the computational studies. Table 3 lists the parameter setting used in the algorithm for all runs.

Table 3. Parameters values used by the proposed algorithm for all runs.

	Value		
Parameter	**Problem 1**	**Problem2**	**Problem3**
Problem size	3×4	7×8	4×5
Number of objective	2	2	3
Population size	20	40	20
Mutation rate	0.50	0.6	0.40
Crossover rate	0.98	0.98	0.98
Maximum generation	200	400	200
Relative tolerance ε	10e-6	10e-6	10e-6

Problem 1: Let us consider the following numerical example presented by many researchers [1,2,3,4,17] to illustrate the application of the proposed algorithm. The problem has the following characteristics:

Supplies: $a_1 = 8,\ \mathrm{a}_2 = 19,\ \mathrm{a}_3 = 17.$

Demands: $b_1 = 11,\ \mathrm{b}_2 = 3,\ \mathrm{b}_3 = 14,\ \mathrm{b}_4 = 16.$

Penalties:

$$C^1 = \begin{bmatrix} 1 & 2 & 7 & 7 \\ 1 & 9 & 3 & 4 \\ 8 & 9 & 4 & 6 \end{bmatrix},\quad \mathrm{C}^2 = \begin{bmatrix} 4 & 4 & 3 & 4 \\ 5 & 8 & 9 & 10 \\ 6 & 2 & 5 & 1 \end{bmatrix}$$

To evaluate the performance of the suggested approach, let us consider the solution of the problem by using different methods. The interactive approach in [17] provides the following results: Z1 =186 and Z2 =174, the fuzzy approach in [2] gave the following results: Z1 = 170 and Z2 = 190, the fuzzy approach in [4] gave the following results: Z(1) = 160 and Z2 = 195, the IFGP approach in [1] gave the following results: Z1 =168 and Z2=185. Also, the obtained result from t-GA in [7] are (143,265), (156,200), (176,175), (186,171), (208,167) with extreme points (143,265) and (208,167).

The proposed approach in this study gave a set of solutions as in figure 7. Figure 7 shows that the results obtained by the proposed algorithm dominate all other results obtained by other different approaches.

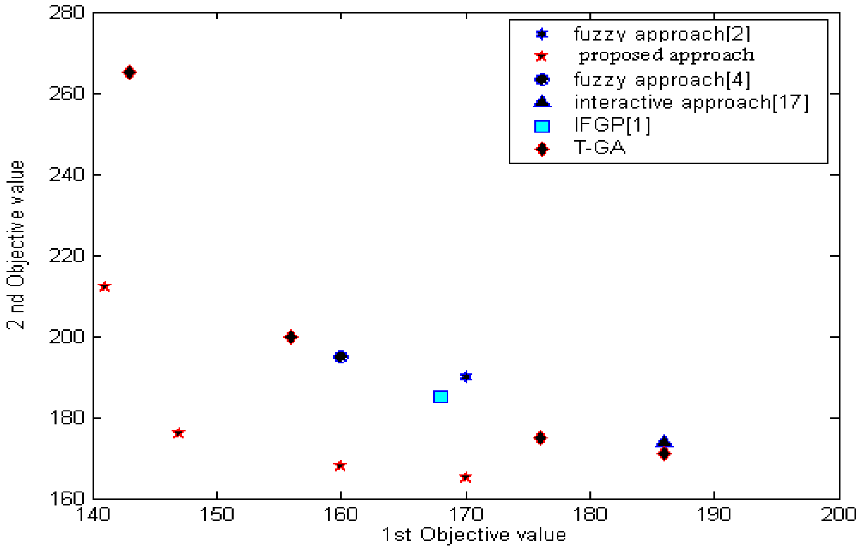


Fig. 7. Comparison between the proposed algorithm and different approaches for first problem.

Problem 2:

Let us consider the following numerical example presented in [7]. The problem has the following characteristics:

Supplies: $a_1 = 10,\ \mathrm{a}_2 = 8,\ \mathrm{a}_3 = 12, a_4 = 16,\ \mathrm{a}_5 = 21,\ \mathrm{a}_6 = 15,\ \mathrm{a}_7 = 7.$

Demand: $b_1 = 9,\ \mathrm{b}_2 = 7,\ \mathrm{b}_3 = 15, b_4 = 10,\ \mathrm{b}_5 = 13,\ \mathrm{b}_6 = 16,\ \mathrm{b}_7 = 7,\ \mathrm{b}_8 = 12.$

Penalties:

$$C^1 = \begin{bmatrix} 1 & 2 & 7 & 7 & 8 & 10 & 9 & 2 \\ 1 & 9 & 3 & 4 & 3 & 5 & 7 & 1 \\ 8 & 9 & 4 & 6 & 4 & 1 & 6 & 9 \\ 2 & 4 & 5 & 5 & 3 & 2 & 3 & 2 \\ 5 & 4 & 5 & 1 & 9 & 9 & 1 & 6 \\ 8 & 3 & 3 & 2 & 2 & 3 & 6 & 7 \\ 1 & 2 & 6 & 4 & 5 & 9 & 3 & 5 \end{bmatrix} \quad C^2 = \begin{bmatrix} 4 & 4 & 3 & 4 & 5 & 8 & 9 & 10 \\ 6 & 2 & 5 & 1 & 7 & 4 & 12 & 4 \\ 2 & 9 & 1 & 8 & 9 & 1 & 4 & 0 \\ 3 & 5 & 5 & 3 & 2 & 8 & 3 & 3 \\ 1 & 4 & 12 & 2 & 1 & 5 & 4 & 9 \\ 2 & 23 & 4 & 4 & 6 & 2 & 4 & 6 \\ 1 & 2 & 1 & 9 & 0 & 13 & 2 & 3 \end{bmatrix}$$

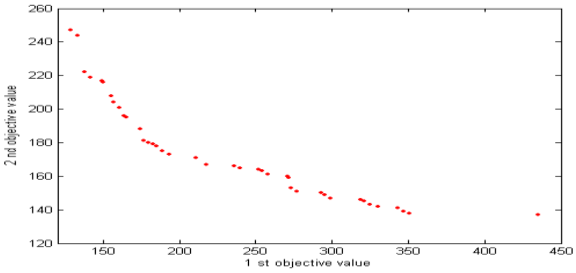


Fig. 8. Comparison between the proposed algorithm and t-GA for second problem

To evaluate the performance of the proposed algorithm, let us consider the solution of the problem by using different methods. The m-GA and t-GA approaches in [7] provide the Pareto set with extreme points (164,359) and (391,164). The proposed algorithm approach in this study has a Pareto set of solution with extreme points (129,247) and (435,137) as in figure 8. Figure 8 shows that the obtained results by the proposed algorithm dominate all the results obtained by the t-GA and m-GA.

Problem 3:

Let us consider the following numerical example presented in [2,17]. The problem has the following characteristics:

Supplies: $a_1 = 5,\ \mathrm{a}_2 = 4,\ \mathrm{a}_3 = 2, a_4 = 9.$

Demand: $b_1 = 4,\ \mathrm{b}_2 = 4,\ \mathrm{b}_3 = 6, b_4 = 2,\ \mathrm{b}_5 = 4.$

Penalties

$$C^1 = \begin{bmatrix} 9 & 12 & 9 & 6 & 9 \\ 7 & 3 & 7 & 7 & 5 \\ 6 & 5 & 9 & 11 & 3 \\ 6 & 8 & 11 & 2 & 2 \end{bmatrix} \quad C^2 = \begin{bmatrix} 2 & 9 & 8 & 1 & 4 \\ 1 & 9 & 9 & 5 & 2 \\ 8 & 1 & 8 & 4 & 5 \\ 2 & 8 & 6 & 9 & 8 \end{bmatrix} \quad C^3 = \begin{bmatrix} 2 & 4 & 6 & 3 & 6 \\ 4 & 8 & 4 & 9 & 2 \\ 5 & 3 & 5 & 3 & 6 \\ 6 & 9 & 6 & 3 & 1 \end{bmatrix}$$

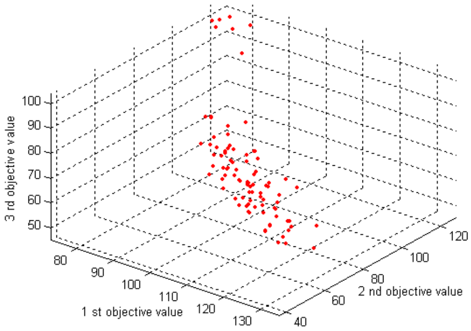


Fig. 9. The obtained solutions by the proposed algorithm for the third Problem.

To evaluate the performance of the suggested approach, let us consider the solution of the problem by using different methods. The fuzzy approach in [2] provides the following results: Z1 =112,z2=106 and Z3 =80. On the other hand, the iterative approach in [17] gave the following results: Z1 = 127,z2=104 and Z3 = 76.
Figure 9 shows the obtained Pareto frontier, The obtained results by the proposed algorithm dominate the results obtained by The fuzzy approach in [2] and the iterative approach in [17].

6. Conclusions

In this paper, we proposed an improved algorithm for solving multiobjective transportation problem MOTP. Our approach has two characteristic features. Firstly, new chromosome's structure was introduced, which is adopted as it is capable to representing all possible feasible solutions. Secondly, the algorithm is an iterative multiobjective genetic algorithm with an external population of Pareto optimal solutions that best conform a Pareto front.
To avoid an overwhelming number of solutions an epsilon dominance process saves the most representative solutions, which gets iteratively updated in the presence of new solutions based on the concept of ε -dominance. The main features of the proposed algorithm could be summarized as follows:

(a) The proposed approach has been effectively applied to solve the MOTP, with no limitation in handing higher dimensional problems.
(b) Allowing a decision maker to control the resolution of the Pareto set approximation by choosing an appropriate ε value according his needs.
(c) The non-dominated solutions in the obtained Pareto-optimal set are well distributed and have satisfactory diversity characteristics.
(d) Low computational time where, the computational time grows with the number of archived solutions.
(e) Simulation results verified the validity and the advantages of the proposed approach.

References

[1] W. F. Abd El-Wahed, M. Sang Lee, Interactive fuzzy goal programming for multiobjective transportation problems. OMEGA ,34(2):158-166, April2006.

[2] W. F. Abd El-Wahed, A multi-objective transportation problem under fuzziness. Fuzzy Sets and Systems 2001;117:27–33.

[3] Y. P. Aneja, K.P.K. Nair, Bicriteria transportation problems. Management Science 1979;25:73–80.

[4] A. K. Bit, M. P. Biswal, S. S. Alam, Fuzzy programming approach to multicriteria decision making transportation problem. Fuzzy Sets and Systems 1992;50:35–41.

[5] D. E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning. Addison Wesley Publishing Company,1989.

[6] M. Gen, Yinzhen L. I, Kenichi I, D.A., solving Multiobjective transportation problem by spanning tree-based genetic algorithm", IEICE Trans. Fundamentals vol.,E82-A,PP 2802-2810,December 1999.

[7] M. Gen, K.Ida E. Kono ad Y. Z. Li, solving bi-criteria solid transportation problem by genetic algorithm," proceeding of the 16th international Conference on computers& industrial Engineering, pp572-575, 1994.

[8] M. Laumanns, L. Thiele, K. Deb, and E. Zitzler, Archiving with guaranteed convergence and diversity in multi-objective optimization.", In GECCO 2002: Proceedings of the Genetic and Evolutionary Computation Conference, Morgan Kaufmann Publishers, New York, NY, USA, p.p. 439-447, July.

[9] T. Lust, A.Jaszkiewicz," Speed-up techniques for solving large-scale biobjective TSP", Computers &OperationsResearch, 37(2010)521—533.

[10] S. M. Lee and L. J. Moore, Optimizing transportation problems with multiple objectives," AIIE transactions , vol.5, pp.333-338,1973.

[11] Z. Michalewicz, Genetic Algorithms + Data Structures = Evolution Programs. Springer-Verlag,3 Rd Edition, 1996.

[12] Z. Michalewicz, G. A. Vignaux, and M. Hobbs, a non standard genetic algorithm for the nonlinear transportation problem, "ORSA Journal on computing, vol.3, no.4 pp.307-316,1991.

[13] K. Miettinen, Non-linear multiobjective optimization. Dordrecht: Kluwer Academic Publisher, 2002.

[14] A. A. Mousa, "Using Genetic Algorithm and Topsis Technique for Multiobjective Transportation Problem: A hybrid approach”, International journal of computer mathematics, under press,2010.

[15] M. S. Osman, M.A. Abo-Sinna, and A. A. Mousa, IT-CEMOP: An Iterative Co-evolutionary Algorithm for Multiobjective Optimization Problem with Nonlinear Constraints. Journal of Applied Mathematics & Computation (AMC), vol.183, pp. 373-389, 2006.

[16] M. S. Osman , M. A. Abo-Sinna , and A. A. Mousa , An Effective Genetic Algorithm Approach to Multiobjective Routing Problems (MORPs).Journal Of Applied Mathematics & Computation (AMC) vol.163. No. (2), 15 April (2005) pp 769-781.

[17] J. L. Ringuest, D. B. Rinks, Interactive solutions for the linear multiobjective transportation problems. European Journal of Operational Research 1987;vol.32: pp96–106.

[18] R. Steuer, Multiple criteria optimization: theory, computation, and application. New York: Wiley; 1986.

[19] G. A. Vignaux, Z. Michalewicz, a genetic algorithm for the linear transpiration problem ,IEEE Trans. Syst., Man. & cybern., vol.21, no3.,pp445-452,1991.

[20] L. Yang, Y. Feng, A bicriteria solid transportation problem with fixed charge under stochastic environment, Applied Mathematics Modeling(2006), doi:10.1016/j.apm. 2006.10.011.