

# Efficient Multiobjective Genetic Algorithm for Solving Transportation, Assignment, and Transshipment Problems

Sayed A. Zaki, Abd Allah A. Mousa, Hamdy M. Genedi, Adel Y. Elmekawy

Department of Mathematic and Statistics, Faculty of Sciences, Taif University, Taif, Saudi Arabia

Email: a\_mousa15@yahoo.com

Received November 21, 2011; revised December 22, 2011; accepted December 30, 2011

## ABSTRACT

This paper presents an efficient genetic algorithm for solving multiobjective transportation problem, assignment, and transshipment Problems. The proposed approach integrates the merits of both genetic algorithm (GA) and local search (LS) scheme. The algorithm maintains a finite-sized archive of non-dominated solutions which gets iteratively updated in the presence of new solutions based on clustering algorithm. The use clustering algorithm makes the algorithms practical by allowing a decision maker to control the resolution of the Pareto set approximation. To increase GAs' problem solution power, local search technique is implemented as neighborhood search engine where it intends to explore the less-crowded area in the current archive to possibly obtain more nondominated solutions. The inclusion of local search and clustering algorithm speeds-up the search process and also helps in obtaining a fine-grained value for the objective functions. 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.

**Keywords:** Transportation Problem; Genetic Algorithms; Local Search; Cluster Algorithm

## 1. Introduction

The transportation problem is considered a special problem of resource allocation, which 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 [1-8].

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 suffi-

cient time, can find solutions as close to the real optimum as we wish. Michalewicz *et al.* [5,6] firstly discussed the use of genetic algorithm (GA) for solving linear and non-linear 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.* [7] 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.* [8] 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.

Evolutionary algorithms suffer from the large size problem of the Pareto set e.g. [9]. 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 [10]. Also evolutionary algorithms such as, genetic algorithms (GAs) can be used as a global optimization tool

for continuous and discrete functions problems. However, a simple GA may suffer from slow convergence, and instability of results [11,12]. GAs' problem solution power can be increased by local searching. In this study a new local random search algorithm in order to reach a quick and closer result to the optimum solution. Local search techniques have long been used to attack many recent optimization problems [13-15]. The basic idea is to start from an initial solution and to search for successive improvements by examining neighboring solutions. The proposed local search technique is based on a dynamic version of pattern search technique. Pattern search technique is a popular paradigm in Direct Search (DS) methods [16].

In this paper we present an improved genetic algorithm to solve MOTP, The algorithm is an iterative multiobjective algorithm with an external population of Pareto optimal solutions that best conform a Pareto Front. Also, GAs' problem solution power can be increased by local searching, where we present a new local random search algorithm in order to reach a quick and closer result to the optimum solution. The remainder of the paper is organized as follows. This paper is organized as follows; Preliminaries is 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. Preliminaries

A general multiobjective optimization problem is expressed by [17]:

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

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

**Definition 1.** [10] (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,

$$f_j(x) \leq f_j(x^*) \text{ for all } j=1, 2, \dots, m \text{ and}$$

$$f_j(x) < f_j(x^*) \text{ for at least one objective function.}$$

**Definition 2.** Clustering algorithm [18].

Let us describe the clustering algorithm which reduces the size of the external population with size  $\bar{N}$  to  $N$  (where  $\bar{N} > N$ ) The clustering approach forms  $N$  clusters from  $\bar{N}$  population by initially assuming each of

$\bar{N}$  members to be a separate cluster, thereafter all  $\binom{\bar{N}}{2}$

Euclidean distances in the objective space are computed. Then, the two cluster with the smallest distance are merged together to form one bigger cluster. This process reduces the number of cluster to  $\bar{N} - 1$ . The inter-cluster distances are computed again and another merging is done. This process is repeated until the number of clusters is reduced to  $N$ . With multiple population member occupying two clusters, the average distance of all pair-wise distances between solutions of the two clusters is used. **Figure 1** illustrates this procedure.

This is especially important in higher dimensional objective spaces, where the clustering algorithm can reduce the required number of solutions considerably. Also, it makes the algorithms practical by allowing a decision maker to control the resolution of the Pareto set approximation by choosing an appropriate clusters number.

## 3. Multiobjective Transportation Problem

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:

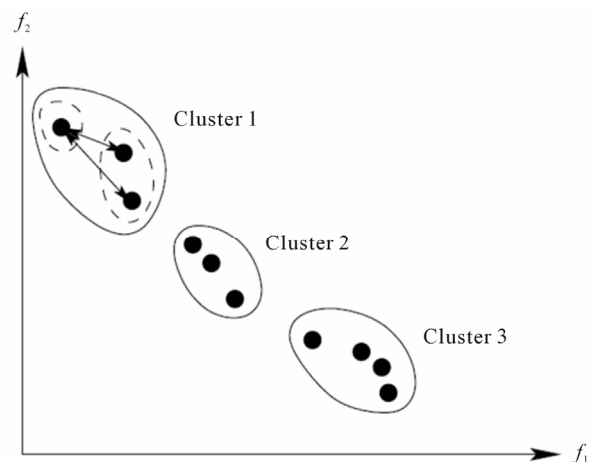
$$\text{MOTP: } F^k(x) = \sum_{i=1}^m \sum_{j=1}^n C_{ij}^k x_{ij},$$

Subject to

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

$$\sum_{i=1}^m x_{ij} = b_j, j = 1, 2, \dots, m$$

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



**Figure 1.** The clustering algorithm.

where  $F^k(x) = \{F^1(x), F^2(x), \dots, F^k(x)\}$  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 ( $k = 1, 2, \dots, K$ ), 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 equal 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 [11,12,19] 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. In the follow-

ing 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 (Figure 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  $i$ . 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 proposed approach.

#### 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 [20].

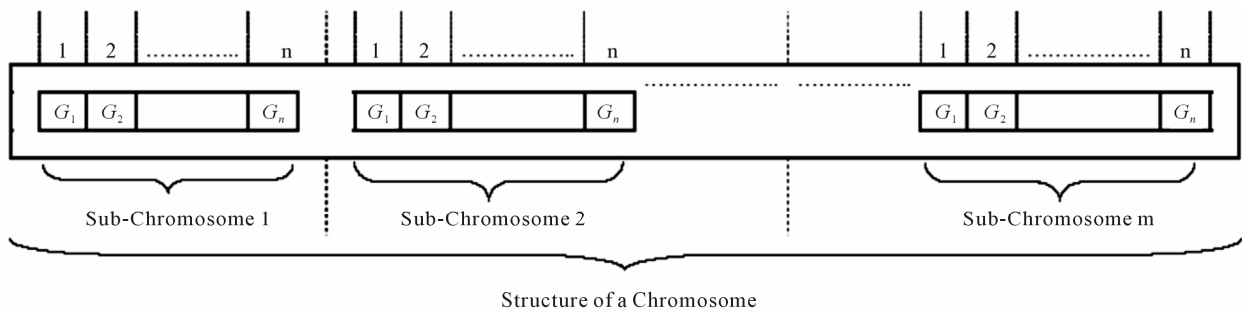


Figure 2. Structure of chromosome for MOTP with  $n$  sources and  $m$  destinations.

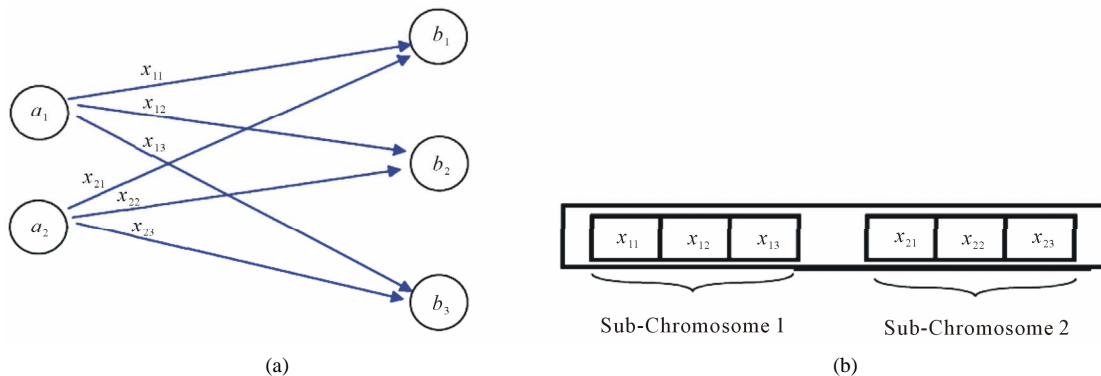


Figure 3. Illustration of chromosome's representation. (a) Transportation graph; (b) Chromosome structure.

**Step 0:** Begin with  $i = 1$ .

**Step 1:** For all  $j = 1, 2, \dots, N$  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 [20] 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:

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

where  $x$  is a string (i.e., individual),  $f(x)$  is a combined fitness function,  $f_i(x)$  is the  $i$ th objective 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  $\psi$  and

$$f_{\min}(\psi) = \min \{f(x) \mid x \in \psi\}$$

### 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**.

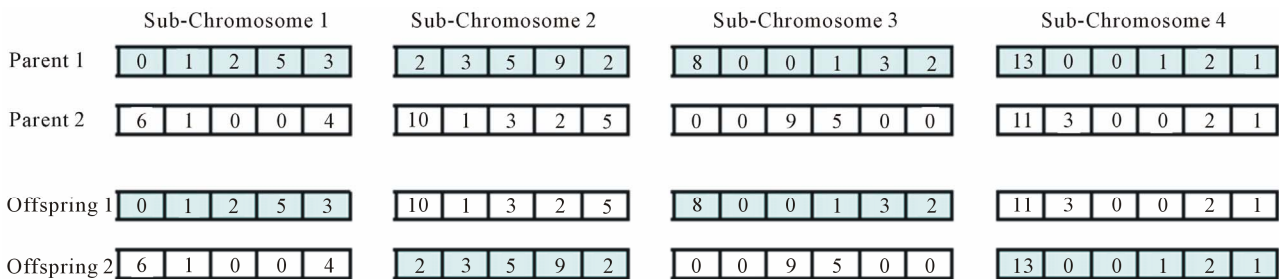
In the example problem in **Figure 4** we have four supplies (that is, we have four sub-chromosomes). The second and fourth sub-chromosome are exchanged between parents. It is interesting here to note that all offspring's chromosome are feasible.

### 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  $i$ th sub-chromosome and then replace it with a random integer within the interval of  $[0, a_i]$ , all other genes in  $i$ th sub-chromosome are generated such that the sum of all genes in the  $i$ th sub-chromosome equal to the  $i$ th supply  $\sum_{j=1}^n Gene_{ij} = a_i$ , as shown in **Figure 5**.

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 1<sup>st</sup> sub chromosome are generated such that the sum of all genes equal to the supply amount

$$\sum_{j=1}^5 G_j = a_1 \Rightarrow \sum G_1 + G_2 + G_3 + G_4 + G_5 = 11.$$



**Figure 4.** Graphs visualizing the crossover operators.

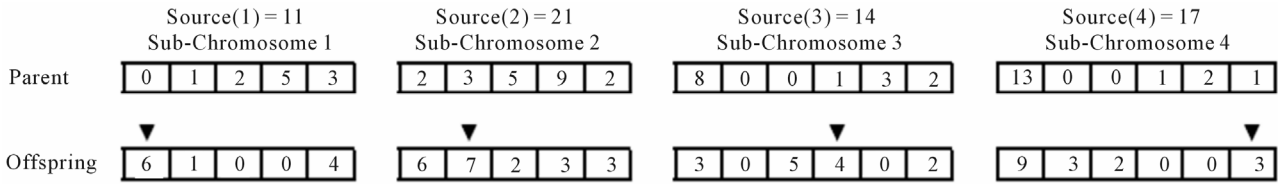


Figure 5. Graphs visualizing the mutation operators.

Through this mutation operator, the population’s feasibility was preserved.

### 4.6. Update Function

Algorithm 1 (Figure 6) 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  $A^{(t)}$ . Also, the function LS is to explore the less-crowded area in the current archive to possibly obtain more nondominated solutions.

As a result the proposed algorithm which is based on the (GA) uses a finite memory, successively updates a finite subset of vectors that 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 to the cluster algorithm Accordingly the algorithm is more practical where a decision maker is able to control the resolution of the Pareto set approximation according to his needs. Also it guarantees an optimal distribution of solutions [9]. The algorithm has a low computational time where, the computational time grows with the number of archived solutions. The proposed algorithm is capable to consider many objective functions. Accordingly it provides the facility to consider more criteria in MOTP problem.

- 
1.  $t = 0$
  2. Create  $P^{(0)}, R^{(0)}$
  3.  $A^{(0)} = \text{nondominated}(P^{(0)})$
  3. while terminate  $(A^{(t)}, t) = \text{false}$  do
  4.  $t := t + 1$
  5.  $P^{(t)} := \text{generate}(A^{(t-1)}, P^{(t-1)})$  {generate new search point}
  6.  $A^{(t)} := \text{update}(A^{(t-1)}, P^{(t)})$  {update archive (algorithm 2)}
  7. end while
  8.  $A := \text{LS}(A)$
  9. Output :  $A^{(t)}$
- 

Figure 6. Algorithm 1.

### 4.7. The Local Search

The local search phase is implemented as a dynamic version of pattern search technique. This study examines the usefulness of a dynamic version of pattern search technique [21] to improve the solution quality of MOTPs. The search procedure looks for the best solution “near” another solution by repeatedly making small changes to a starting solution until no further improved solutions can be found.

The local search is started by loading the Pareto solutions in the external archive  $A$ . For every solution  $x_m$  in the archive, we have  $(X_m \in R^n) \in A$ , where the changes on the values for each dimension  $(i = 1, 2, \dots, n)$  can be defined as

$$\Delta(x_m^k) = R \left( 1 - r^{\left( \frac{1-k}{k_{\max}} \right)} \right)$$

where  $k$  is number of trial  $(k = 0, \dots, k_{\max})$  to obtain preferred solution than  $X_m$ ,  $r$  is the random number in the range  $[0,1]$ ,  $R$  is the search radius.

We successively look at the points  $x_+ = x_m \pm \Delta(x_m^k) e_i$ ,  $(i = 1, 2, \dots, n)$ , until we find  $x_+$  for which  $f_j(x_+) > f_j(x_t)$  for at least one objective. If we find no  $x_+$  such that  $f_j(x_+) > f_j(x_t)$ , then  $x_+ = x_m$ . Then we update the Pareto solutions by non dominated ones and the dominated ones are removed. This situation is represented in Figure 7 for the case in  $R^2$ . Without

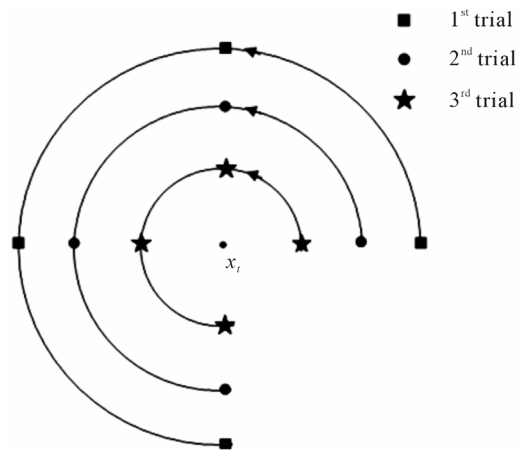


Figure 7. Mechanism of dynamic pattern search in  $R^2$ .

loss of generality, the elements discussed above are synthesized to evolve the proposed approach. The pseudo code of the proposed algorithm is given in **Figure 8**.

### 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 1** lists the parameter setting used in the algorithm for all runs. Let us consider the following numerical example presented by many researchers [3,4,22,23,24] to illustrate the application of the proposed algorithm.

**Problem 1:**

The problem has the following characteristics:

Supplies:  $a_1 = 8, a_2 = 19, a_3 = 17$ .

Demands:  $b_1 = 11, b_2 = 3, b_3 = 14, b_4 = 16$ .

Penalties:

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

To evaluate the performance of the suggested ap-

**Local search phase by adaptive pattern search**

Input :  $A$ , search radius

$k = 0$ , load the Pareto solutions from the external archive

**while**  $k \leq k_{max}$  **do**

**for**  $j = 1:n$  **do**

$x_+ = x_m \pm \Delta(x_m^k)e_i, (i = 1, \dots, n)$

**If**  $x_+ > x_m$  **do**

$x_m = x_+$  {replace the old solution}

**end if**

**end for**,  $k = k + 1$

**end while**

Output  $\rightarrow A$

**Figure 8.** The pseudo code of the proposed approach.

**Table 1.** Parameters values used by the proposed algorithm for all runs.

Value		Parameter
Problem 2	Problem 1	
4×5	3×4	Problem size
3	2	Number of objective
15	15	Population size
0.5	0.50	Mutation rate
0.95	0.95	Crossover rate
100	100	Maximum generation

proach, let us consider the solution of the problem by using different methods. The interactive approach in [24] provides the following results: Z1 = 186 and Z2 = 174, the fuzzy approach in [4] gave the following results: Z1 = 170 and Z2 = 190, the fuzzy approach in [23] gave the following results: Z1 = 160 and Z2 = 195, the IFGP approach in [3] 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). The proposed approach in this study gave a set of solutions as in **Figure 9**. It is obviously that the results obtained by the proposed algorithm dominate all other results obtained by other different approaches.

**Problem 2:**

Let us consider the following numerical example presented in [3,4]. The problem has the following characteristics:

Supplies:  $a_1 = 5, a_2 = 4, a_3 = 2, a_4 = 9$ .

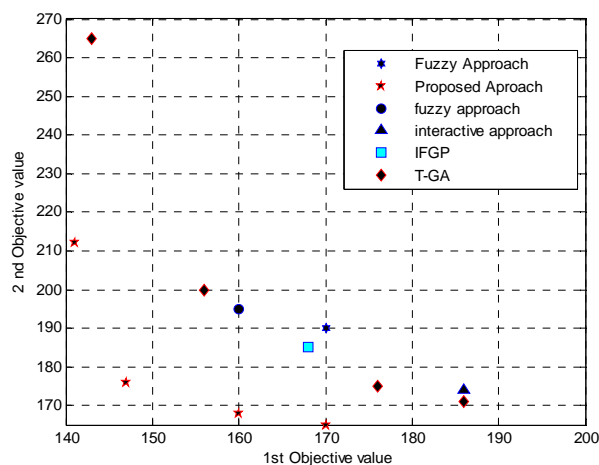
Demand:  $b_1 = 4, b_2 = 4, b_3 = 6, b_4 = 2, b_5 = 4$ .

Penalties

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

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

To evaluate the performance of the suggested approach, let us consider the solution of the problem by using different methods. The fuzzy approach in [4] provides the following results: Z1 = 112, Z2 = 106 and Z3 = 80. On the other hand, the interactive approach in [3]



**Figure 9.** Comparison between the proposed algorithm and different approaches for first problem.



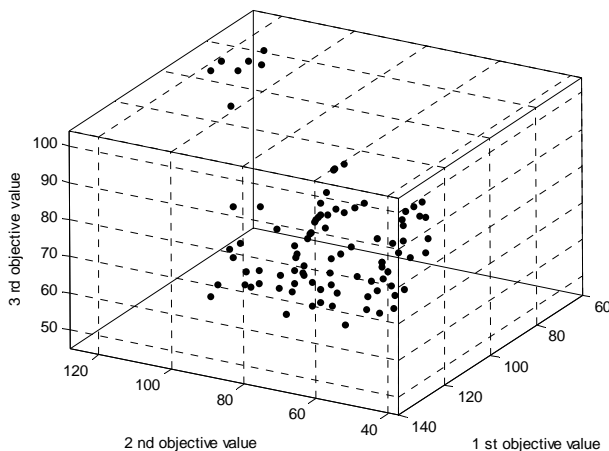
gave the following results:  $Z1 = 127$ ,  $Z2 = 104$  and  $Z3 = 76$ .

**Figure 10** shows the obtained Pareto frontier, The obtained results by the proposed algorithm dominate the results obtained by The fuzzy approach in [4] and the interactive approach in [3]. From the previous result, it was concluded that Integration of GA and local search technique has improved the quality of the founded solution, Also it guarantee the faster converge to the Pareto optimal solution. GA has provided the initial set (close to the Pareto set as possible) followed by local search method to improve the quality of the solutions. However, because of its stochastic behavior, GA may suffer from slow convergence. In order to reach a quick and closer result to Pareto optimal solution, and to improve the efficiency of the GA, the algorithm maintain an external archive of the observed nondominated solutions that best conform a Pareto Front.

## 6. Conclusions

In this paper, an improved algorithm for solving MOTP was presented. Our approach has two characteristic features. Firstly, the algorithm is an iterative multiobjective genetic algorithm with an external population of Pareto optimal solutions that best conform a Pareto front. Secondly the algorithm implements GA to provide the initial set (close to the Pareto set as possible) followed by local search method to improve the quality of the solutions. It is concluded that Integration of GA and local search technique has improved the solution's quality. To avoid an overwhelming number of solutions clustering algorithm saves the most representative solutions, which gets iteratively updated in the presence of new solutions. The main features of the proposed algorithm could be summarized as follows:

1) The proposed approach has been effectively applied



**Figure 10.** Pareto set using the proposed algorithm for the third problem.

to solve the MOTP, with no limitation in handling higher dimensional problems.

2) The proposed algorithm was able to find well distributed of the Pareto-optimal curve in the objective space.

3) The proposed algorithm keeps track of all the feasible solutions found during the optimization and therefore do not have any restrictions on the number of the Pareto-optimal solutions found.

4) The inclusion of local search speeds-up the search process and also helps in obtaining a fine-grained value for the objective functions.

5) The success of our approach on most of the test problems not only provides confidence but also stress the importance of hybrid evolutionary algorithms in solving multiobjective optimization problems.

## REFERENCES

- [1] A. A. Mousa, "Using Genetic Algorithm and TOPSIS Technique for Multiobjective Transportation Problem: A Hybrid Approach," *International Journal of Computer Mathematics*, Vol. 87, No. 13, 2010, pp. 3017-3029. [doi:10.1080/00207160902875262](https://doi.org/10.1080/00207160902875262)
- [2] L. Yang and Y. Feng, "A Bicriteria Solid Transportation Problem with Fixed Charge under Stochastic Environment," *Applied Mathematical Modelling*, Vol. 31, No. 12, 2007, pp. 2668-2683. [doi:10.1016/j.apm.2006.10.011](https://doi.org/10.1016/j.apm.2006.10.011)
- [3] W. F. Abd El-Wahed and S. M. Lee, "Interactive Fuzzy Goal Programming for Multiobjective Transportation Problems," *Omega*, Vol. 34, No. 2, 2006, pp. 158-166. [doi:10.1016/j.omega.2004.08.006](https://doi.org/10.1016/j.omega.2004.08.006)
- [4] W. F. Abd El-Wahed, "A Multi-Objective Transportation Problem under Fuzziness," *Fuzzy Sets and Systems*, Vol. 117, No. 1, 2001, pp. 27-33. [doi:10.1016/S0165-0114\(98\)00155-9](https://doi.org/10.1016/S0165-0114(98)00155-9)
- [5] Z. Michalewicz, G. A. Vignaux and M. Hobbs, "a Non-standard Genetic Algorithm for the Nonlinear Transportation Problem," *INFORSA Journal on Computing*, Vol. 3, No. 4, 1991, pp. 307-316.
- [6] G. A. Vignaux and Z. Michalewicz, "a Genetic Algorithm for the Linear Transportation Problem," *IEEE Transactions on Systems, Man & Cybernetics*, Vol. 21, No. 3, 1991, pp. 445-452. [doi:10.1109/21.87092](https://doi.org/10.1109/21.87092)
- [7] M. Gen, K. Ida Kono and Y. Z. Li, "Solving Bi-Criteria Solid Transportation Problem by Genetic Algorithm," *Proceeding of the 16th International Conference on Computers & Industrial Engineering*, San Antonio, 2-5 October 1994, pp. 572-575.
- [8] M. Gen, Y. Z. Li and Kenichi Ida, "Solving Multiobjective Transportation Problem by Spanning Tree-Based Genetic Algorithm," *IEICE Transactions on Fundamentals*, Vol. E82-A, No. 2, 1999, pp. 2802-2810.
- [9] M. Laumanns, L. Thiele, K. Deb and E. Zitzler, "Archiving with Guaranteed Convergence and Diversity in Multi-Objective Optimization," *Proceedings of the Genetic and Evolutionary Computation Conference*, New York,

- 9-13 July 2002, pp. 439-447.
- [10] 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*, Vol. 183, No. 1, 2006, pp. 373-389.
- [11] Z. Michalewicz, "Genetic Algorithms + Data Structures = Evolution Programs," 3rd Edition, Springer-Verlag, Berlin, 1996.
- [12] M. Gen and R. Cheng, "Genetic Algorithms & Engineering Design," John Wiley & Sons, New York, 1997.
- [13] Y. Kilani, "Comparing the Performance of the Genetic and Local Search Algorithms for Solving the Satisfiability Problems," *Applied Soft Computing*, Vol. 10, No. 1, 2010, pp. 198-207. [doi:10.1016/j.asoc.2009.07.012](https://doi.org/10.1016/j.asoc.2009.07.012)
- [14] G. Whittaker, et al., "A Hybrid Genetic Algorithm for Multiobjective Problems with Activity analysis-Based Local Search," *European Journal of Operational Research*, Vol. 193, No. 1, 2009, pp. 195-203. [doi:10.1016/j.ejor.2007.10.050](https://doi.org/10.1016/j.ejor.2007.10.050)
- [15] C. Hamzacebi, "Improving Genetic Algorithms' Performance by Local Search for Continuous Function Optimization", *Applied Mathematics and Computation*, Vol. 196, No. 1, 2008, pp. 309-317. [doi:10.1016/j.amc.2007.05.068](https://doi.org/10.1016/j.amc.2007.05.068)
- [16] A. A. Mousa, R. M. Rizk-Allah and W. F. Abd El-Wahed, "A Hybrid Ant Colony Optimization Approach Based Local Search Scheme Formultiobjective Design Optimizations," *Electric Power Systems Research*, Vol. 81, No. 4, 2011, pp. 1014-1023. [doi:10.1016/j.epsr.2010.12.005](https://doi.org/10.1016/j.epsr.2010.12.005)
- [17] K. Miettinen, "Non-Linear Multiobjective Optimization," Kluwer Academic Publisher, Dordrecht, 2002.
- [18] M. S. Osman, M. A. Abo-Sinna and A. A. Mousa, "Epsilon-Dominance Based Multiobjective Genetic Algorithm for Economic Emission Load Dispatch Optimization Problem," *Electric Power Systems Research*, Vol. 79, No. 11, 2009, pp. 1561-1567.
- [19] C. M. Fonseca and P. J. Fleming, "An Overview of Evolutionary Algorithms in Multiobjective Optimization," *Evolutionary Computation*, Vol. 3, No. 1, 1995, pp. 1-16. [doi:10.1162/evco.1995.3.1.1](https://doi.org/10.1162/evco.1995.3.1.1)
- [20] 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*, Vol. 163, No. 2, 2005, pp. 769-781.
- [21] S. S. Rao, "Engineering Optimization Theory and Practice," 3rd Edition, John Wiley, New York, 1996, p. 903.
- [22] Y. P. Aneja and K. P. K. Nair, "Bicriteria Transportation Problems," *Management Science*, Vol. 25, No. 1, 1979, pp. 73-80. [doi:10.1287/mnsc.25.1.73](https://doi.org/10.1287/mnsc.25.1.73)
- [23] A. K. Bit, M. P. Biswal, S. S. Alam, "Fuzzy Programming Approach to Multicriteria Decision Making Transportation Problem," *Fuzzy Sets and Systems*, Vol. 50, No. 2, 1992, pp. 35-41. [doi:10.1016/0165-0114\(92\)90212-M](https://doi.org/10.1016/0165-0114(92)90212-M)
- [24] J. L. Ringuest and D. B. Rinks, "Interactive Solutions for the Linear Multiobjective Transportation Problems," *European Journal of Operational Research*, Vol. 32, No. 1, 1987, pp. 96-106. [doi:10.1016/0377-2217\(87\)90274-8](https://doi.org/10.1016/0377-2217(87)90274-8)