Palestine Journal of Mathematics

Vol. 11 (Special Issue III)(2022) , 189–201

VARIANTS OF GENETIC ALGORITHM TO SOLVE MULTI-OBJECTIVE INTERVAL SOLID TRANSPORTATION PROBLEM

Shubha Agnihotri and Jayesh M. Dhodiya

Communicated by Harikrishnan Panackal

MSC 2010 Classifications: Primary 90B06; Secondary 90B50.

Keywords and phrases: Multi-objective interval solid transportation problem, GA , NSGA II, NSGA III.

Abstract This paper addresses the structure of the multi-objective interval solid transportation problem (MOISTP) within the framework of a hybrid genetic algorithm, non-dominated sorting genetic algorithm (NSGA) II, and NSGA III. Here, the solid transportation problems (STP) are specific transportation problems wherein the mode of transportation is taken into account along with demand and supply constraints. In practice, it is argued that crisp data is hypothetical due to limitations on the measurement scale. To take care of said impreciseness/ uncertainty, the concept of interval or fuzzy numbers is generally employed. In this paper, the authors cope with uncertainty with intervals and propose a matrix-based stochastic algorithm to generate an initial population to apply variants of genetic algorithm to solve MOISTP. In this connection, a numerical illustration is also posed and solved using a hybrid genetic algorithm, NSGA II and NSGA III, and obtained results are compared with the fuzzy programming technique (FPT).

1 Introduction

In traditional Transportation Problem, sources have a fixed quantity of transfer capacity and destinations require a specified volume at the same time, which may differ from the capacity of the source. Because of these considerations, the DM's main purpose is to reduce transportation costs while staying within the study's source and demand restrictions. Hitchcock [11] was the first to study its mathematical structure in 1941, and Dantzig [4] offered a solution strategy for the traditional transportation problem using lp ideas. In addition, multi-dimensional objectives, rather than uni-dimensional objectives, have become the order of the day to address rising transportation issues. When compared to one-dimensional goals, multi-dimensional objectives are inherently conflicting, such as lowering transportation costs, improving product quality, increasing user preparedness, and so on. The occurrence of multi-objective scenarios on a wide scale in the actual world has altered the paradigm away from traditional TP and toward multi-objective transportation challenges. Furthermore, in fact, maintaining conveyance homogeneity for shipment is often impossible; as a result, heterogeneous conveyances, such as ships, cargo, trains, or any combination of these, are used. As a result, in order to determine the impact of heterogeneous conveyances, an additional restriction is required. This specific paradigm, dubbed the STP, was first studied by Schell [19]. Furthermore, there is a great deal of imprecision in data in today's business sector. Data is ambiguous, imprecise, and unreliable. As a result, data is considered as interval numbers or fuzzy numbers. When the data is considered as interval numbers, the formed problem is called MOISTP.

Besides, there are many classical approaches, such as fuzzy programming technique, gradient-based method, weighted sum approach, etc., to solve transportation problems. But these methods mostly face some common problems such as dependency on an initial solution, being stuck to a sub-optimal solution, giving a solution to one problem but not being efficient in solving other problems, etc. Moreover, to solve the multi-objective problem, these algorithms convert multiple objectives into a single objective first and then provide a single Pareto-optimal solution. On the other hand, evolutionary algorithms are a class of algorithms that alleviate these drawbacks. An evolutionary algorithm, an iterative and stochastic process, mimic natural evolutionary principal and work on a population of solution and provide multiple Pareto-optimal solutions in a single simulation run. A genetic algorithm, invented by John Holland in 1960-1970 and specifically based on Darwin's theory of "survival of fittest" is an evolutionary algorithm. It starts by employing a population of solutions and, with the help of genetic operators, developing potential solutions.

In multi-objective optimization setting, it is argued that one should obtain as many solutions as possible and these solution should be diverse (see [7]). In this respect, Schaffer [18] has done pioneer work. he developed vector evaluated genetic algorithm(VEGA). But this algorithm satisfies first goal of finding Pareto-optimal solution but fail in acheiving second goal of maintaining diversity between them. Because of that, VEGA is susceptible towards specific part of Pareto-optimal front. After a decade, When Goldberg proposed the use of the notion of non-dominated sorting in evolutionary algorithms, it ushered in a revolution in the field of multi-objective evolutionary algorithms. Following that, researchers devised a variety of algorithms. WBGA ([9]), MOGA by Fonseca [8], NPGA([26]), and NSGA([21]) are among the organisations that attract notice. Furthermore, NSGA is directly based on the Goldberg's suggestion . However, this algorithm's key drawbacks are its lack of elitism, the requirement to supply a user-defined parameter, and its high computational complexity. Deb [7] gave its updated version NSGA II, equipped with elitism and explicit diversity preservation operator, in 2002 to alleviate NSGA's shortcoming. Wang et. al. [22] construct a multi-objective mathematical model for fault diagnosis problem and convert constraints of the model into objective functions by adopting penalty method. Then NSGA II is employed to find the fault diagnosis result. Soyel et.al. [20] proposed a method based on NSGA II for facial expression recognition. Some other applications of NSGA II can be found in [14, 15]. However, because of the crowding distance operator, NSGA II also has a high computing complexity and fails as the number of targets increases. To help with these problems, Deb [6] released NSGA III in 2014, which is mostly based on the reference point technique. Wangsom et. al. [24] employed NSGA III to get solutions for a multi-objective optimization of scheduling on cloud by considering makespan, cost and VM utilizations as objectives. Wang et al. [23] propose a methodology including NSGA III and fuzzy C-means clustering algorithm to the environmental management problem in china's iron and steel industry. He et al.[10] proposed a model for rush order insertion rescheduling problem and solved it by NSGA III. Some other interesting application of NSGA III can be found in [3, 12].

Jimenez et. al. [13] employed GA on MOISTP. Nagarajan et al. [16] considered MOISTP with all the parameters (source, demand, and conveyance capacity) as stochastic intervals numbers and solved it with a fuzzy programming approach. Baidya et al. [2] introduced safety factors in transportation problems and solved MOISTP with safety measures by the reduced gradient method. Baidya et al. [1] employed the weighted tchebycheff method on MOISTP with budget constraints. Nagarajan et al. [17] solved a MOSTP by considering its source and demand parameters as interval numbers.

Since evolutionary algorithms begin with population generation, they can yield multiple solutions in a single simulation run. They also require only objective function values to solve the problem; as a result, they can be used to solve complex problems. Therefore to solve MOISTP, we begin with hybrid GA, NSGA II, and NSGA III. NSGA III also performs effectively with a growing number of objectives while requiring less computational effort. Therefore authors offer a strategy for generating an initial population in view of applying NSGA III on MOISTP. In addition, topsis method is incorporated with NSGA III and NSGA II to get the best feasible solution from the Pareto-optimal set according to the weight provided by DM.

This paper contains five sections. In section 2, preliminary knowledge about the multi-objective problem is given, followed by the mathematical model of MOISTP. In section 3, the solution procedure of the genetic algorithm, hybrid genetic algorithm, NSGA II, NSGA III, and topsis method is discussed. Section 4 consists of a numerical example to show the effectiveness of these methods with sensitivity analysis. The last conclusion is given, followed by references.

2 Preliminaries

This section introduces some basic fundamentals of Interval Theory in the light of multi-objective optimization.

Risk Attitude Parameter:

To convert MOISTP into crisp MOSTP, the risk attitude parameter is utilized. Let $C^q = [c^q_{ijk}]$ is the coefficient matrix corresponding to q^{th} objective. Then C^q converted into its equivalent crisp values by the following formula:

$$C^{\epsilon}_{ijk} = [c^{\epsilon}_{ijk}] = \frac{c^U_{ijk} + c^L_{ijk}}{2} + \epsilon(c^U_{ijk} - c^L_{ijk}) \tag{2.1}$$

for q^{th} objective.
Where ϵ is the risk attitude parameter for the uncertain data assumed by the decision-maker having values in the range of $-.5$ to $.5$. c^U_{ijk} and c^L_{ijk} are upper and lower bound for that specific interval. Risk attitude parameter is optimistic, most likely and pessimistic for $[-.5, 0), 0$ and $(0, .5]$ respectively.

Concept of Domination:

A solution $\vec{x}$ is said to dominate another solution $\vec{y}$ iff:

(i) $f_q(\vec{x}) \leq f_q(\vec{y})$ for all indices $q \in \{1, 2, \cdots Q\}$

(ii) $f_{\bar{q}}(\vec{x}) < f_{\bar{q}}(\vec{y})$ for at least one index $\bar{q} \in \{1, 2, \cdots Q\}$

If there is no other solution $\vec{y}$ in the feasible space that meets the above two conditions, a solution $\vec{x}$ is said to be Pareto-optimal.

3 Mathematical Formulation of MOISTP

The mathematical formulation of MOISTP can be given as follows:
Model 1:

$$\textbf{min } Z_q(x) = \sum_{r=1}^{R} \sum_{s=1}^{S} \sum_{t=1}^{T} [f_{Urst}, f_{Lrst}]^q x_{rst}, \qquad q \geq 2, \tag{3.1}$$

subject to the constraints:

$$\sum_{s=1}^{S} \sum_{t=1}^{T} x_{rst} \in [a_{Ur}, a_{Lr}], \; r = 1(1)R, \tag{3.2}$$

$$\sum_{r=1}^{R} \sum_{t=1}^{T} x_{rst} \in [b_{Us}, b_{Ls}], \; s = 1(1)S, \tag{3.3}$$

$$\sum_{r=1}^{R} \sum_{s=1}^{S} x_{rst} \in [c_{Ut}, c_{Lt}], \; t = 1(1)T, \tag{3.4}$$

$$x_{rst} \geq 0, \forall r, s, t,$$

$$\sum_{r=1}^{R} a_{Ur} \geq \sum_{s=1}^{S} b_{Us}; \sum_{r=1}^{I} a_{Lr} \geq \sum_{s=1}^{J} b_{Ls}; \sum_{t=1}^{T} e_{Ut} \geq \sum_{s=1}^{J} b_{Us}; \sum_{t=1}^{K} e_{Lt} \geq \sum_{s=1}^{J} b_{Ls}. \tag{3.5}$$

where q represent the number of objectives considered. The supply, demand and conveyance capacity is represented as $[a_{Ur}, a_{Lr}], [b_{Us}, b_{Ls}]$ and $[c_{Ut}, c_{Lt}]$ for r^{th} origin, s^{th} destination and t^{th} conveyance respectively. The quantity x_{rst} denotes the unknown amount of goods transported from r^{th} origin to s^{th} destination utilizing t^{th} conveyance.

4 Solution Procedure

4.1 Genetic Algorithm

Genetic algorithms start with the initialization of the population and provide a potential solution by applying genetic operators to the population. The process of GA can be described in the following steps:

(i) The first step in GA is to initialize the population. To initialize the population, we have developed a stochastic matrix-based population approach which is discussed in section (4.5).

(ii) In GA, the initialization process is followed by fitness evaluation. In most cases, it is made equal to the objective function value. However, in NSGA II and NSGA III ranking scheme is used.

(iii) After fitness evaluation, chromosomes undergo the selection operator, and better chromosomes are selected for the crossover operator.

(iv) In the crossover, two chromosomes are selected for the crossover, which produces a child population. In our work, we have used an arithmetic crossover operator.

(v) The last step of the genetic operator is mutation. The mutation is done to avoid a trap in the local minima and maintain diversity among the population.

(vi) Steps (ii) to (v) are repeated until termination criteria are met. In our work predefined maximum number of generations is the termination criteria.

4.2 Hybrid Genetic Algorithm

To highlight the multiple value judgment and complex dynamic changes that occurred in the process of decision making, Dhodiya and Tailor([25]) incorporated the concept of aspiration level(AL) into GA. They convert the multi-objective problem into the single objective problem by aspiration level and then solve it by GA. The stepwise procedure is given in the following steps:

(i) Consider the mathematical model of MOISTP and convert the model into the crisp model by using the risk attitude parameter.

(ii) Find the positive ideal solution (PIS) and negative ideal solution (NIS) for each of the objectives of the crisp model and evaluate exponential membership value for each of them.

$$\mu_{f_q(x)} = \begin{cases} 1; & \text{if} f_q \leq f_q^{PIS}, \\ \frac{\exp(-S*\psi_q(x)) - \exp(-S)}{1-\exp(-S)}; & \text{if} f_q^{PIS} \leq f_q \leq f_q^{NIS}, \\ 0; & \text{if} f_q \geq f_q^{NIS}, \end{cases} \tag{4.1}$$

where, $\psi_q(x) = \frac{f_q - f_q^{PIS}}{f_q^{NIS} - f_q^{PIS}}$, and S is non-zero shape parameter(Sp), regulated by the DM and $0 \leq \mu_{f_q}(x) \leq 1$. It should be noted that the membership function in $[f_q^{PIS}, f_q^{NIS}]$ is strictly convex (concave) for $S < 0, (S > 0)$.

(iii) Convert multi-objective problem into single objective problem according to Gupta et al. as follow:
Model-2 objective function:

$$maxW = \prod_{k=1}^{m} \mu_{f_q}, \tag{4.2}$$

Subject to the constraints: (3.2) to (3.4)
and

$$\mu_{f_q}(x) - \overline{\mu_{f_q}(x)} \geq 0; q = 1, 2, \cdots Q. \tag{4.3}$$

Where the required AL of fuzzy goals corresponding to each objective is $\mu f_q(x)$, however, the model mentioned above can be solved by altering the DM's ALs to achieve various fuzzy goals.

(iv) Discover the various transportation schemes for the Model-2, developed in step-(iii), through GA with different choices of the Sp.

4.3 NSGA II

To understand NSGA II, we first understand its two main operators: non-dominated sorting procedure and crowded comparison operator.

Non-dominated Sorting

To do the sorting of fronts, we have to calculate two entities:

(i) The non-negative integer, which reflects the number of solutions that the solution p dominate, called as domination count n_p.

(ii) A set of solutions which are dominated by solution p.

Now we start the process by comparing each member of the population to each other using the concept of dominance, which provided the first/primary non-dominated front. All solutions belonging to the main non-dominated front will have a dominance count of zero. Now, for each solution with $n_p = 0$, we walk over each member q of its set S_p and apply the dominance concept to each one, lowering the dominance count by one. Every member with a domination count of zero is moved to a different list during this operation. The second non-dominated front is the label given to a member of this new list. The third front is determined in the same way by repeating this procedure with each member of Sq. This process is carried out until all feasible fronts have been identified.

Crowding Distance Tournament Selection Operator

We take the average distance of two solutions on either side of the solution i along each of the objectives to derive an estimate of the density of solutions surrounding a particular solution i in the population. This statistic d_i is an estimate of the perimeter of the cuboid produced by employing the nearest neighbors as vertices, which is referenced as the crowding distance.

To estimate crowding distance, we first set elements of a particular front in worse order of their objective values and then assign a very high crowding distance infinity to boundary solutions, and for the rest solution, crowding distance is calculated by the following formula:

$$l_{(I_j^q)} = l_{(I_j^q)} + \frac{f_q^{(I_{j+1}^q)} - f_q^{(I_{j-1}^q)}}{f_q^{max} - f_q^{min}}. \tag{4.4}$$

Where I_j is the j^{th} solution in the sorted list . f_q^{max} and f_q^{min} are the maximum and minimum objective values for q^{th} criterion.
A solution i wins the crowding distance tournament with another solution j if it has better rank or if it has same rank then it should have better crowding distance.

Process of NSGA II

This section outlines the NSGA II algorithm in the following two phases. In phase 1, the initial population (A_t) of size N is generated, which is further segregated into different fronts by employing the non-dominated sorting procedure. Next, we have assigned fitness to solutions. Then we applied three genetic operators, viz. selection, crossover, and mutation, and obtained the offspring population (B_t) of size N. In second phase, initial population A_t and offspring population B_t are combined to form population C_t of size $2N$. Then implement the NDS procedure on combined population C_t and get different fronts (such as F_1, F_2 and so on). In order to select N members, we first choose fronts according to their ranking and add them into a new empty set S_t and check whether the cardinality of S_t is greater than or equal to N and start adding fronts to S_t until cardinality of S_t is greater than or equal to N. If S_t has exactly N population members, then we apply genetic operators on it and get a new population A_{t+1} else, we apply a crowded distance tournament selection operator to choose $N - \textit{cardinality of} \;\; S_t$ population members then apply genetic operator. This phase continues until the termination criterion is met.

4.4 NSGA III

The key difference between NSGA II and NSGA III lies in the working principle of the selection operator. In NSGA III, to select a partial number of population members from the last front, a reference point strategy is used, which is discussed below:

Step 1 We began the algorithm by forming translated objective function as $f_i^{'}(x) = f_i(x) - Z_i^{min}$ by obtaining $Z_i^{min}(i = 1, 2, \cdots M)$ from $\cup_{\tau=0}^{t} S_\tau$.

Step 2 In this step, normalization of objective functions is done. For this M- dimensional hyperplane is generated with the help of M extreme vectors obtained in the previous step. Then we calculate intercept x_i of the i^{th} axis and normalize objective functions using $f_i^n(x) = \frac{f_i^{'}(x)}{x_i}, \forall i = 1, 2, \cdots M$.

Step 3 Now, By using Das and Dennis [5] approach, reference points are generated.

Step 4 In this step, we create reference lines by linking the reference points and the origin. Then the perpendicular distance between each population member of S_t and the reference line is calculated, and the reference point with the minimum perpendicular distance is considered to be associated with the population member.

Step 5 Finally, the niche preservation operator is used to choose a member from the last front to fulfill the vacant position of A_{t+1}.

Niche Preservation Operator

To begin, calculate the niche count ρ_h for the h^{th} reference point as the number of individuals in S_t/F_l who are linked to the h^{th} reference point. Then look for

$$H_{min} = \{h : argmin_h \rho_h\}.$$

When $|H_{min}| \geq 1$, one $\bar{h} \in H_{min}$ is chosen at random. There are two possibilities now:

(i) If $\rho_{\bar{h}} = 0$, one or more individuals may be associated with $\bar{h}$ in front F_l. In this scenario, the individual with the shortest perpendicular distance to $\bar{h}$ is chosen for the A_{t+1} population, and if there is no member associated with $\bar{h}$, reference point $\bar{h}$ is removed from consideration for the current generation.

(ii) if $\rho_{\bar{h}} \geq 1$ then if there exist individuals in the set F_l that are associated to $\bar{h}$ then any one can be chosen randomly.

each time after addition of new individual to A_{t+1}, $\rho_{\bar{h}}$ incremented by one.
This process is repeated k number of times.

4.5 Topsis Method

The Topsis method developed by Hwang and Yoon chooses the best alternative, which has the shortest Euclidean distance from PIS and farthest from NIS. The steps of the Topsis method are given as follows:

- Suppose we have n alternatives and m attributes, make a table of order $n \times m$ whose enteries are denoted as t_{ij}.
- Normalize the obtained table in step 1 as:

$$N_{ij} = \frac{t_{ij}}{\sum_{j=1}^{m} t_{ij}^{2}{}^{\frac{1}{2}}}$$

- Give the weight to each objective such as $\sum_{j=1}^{m} W_j = 1$.
- Multiply weight to normalize table and get table $R_{ij} = w_j.N_{ij}$.

- Calculate

$$R^{+} = \Big\{\Big(\sum_{i}^{max} R_{ij}|j \in J\Big), \Big(\sum_{i}^{min} R_{ij}|j \in J'\Big)|i = 1(1)n\Big\}$$
$$= \{R_1^{+}, R_2^{+}, \cdots R_m^{+}\}$$
$$R^{-} = \Big\{\Big(\sum_{i}^{min} R_{ij}|j \in J\Big), \Big(\sum_{i}^{max} R_{ij}|j \in J'\Big)|i = 1(1)n\Big\}$$
$$= \{R_1^{-}, R_2^{-}, \cdots R_m^{-}\}$$

  where j refers to beneficial attributes and $J^{'}$ refers to non-beneficial attribute.

- Calculate the euclidean distance of alternatives from the ideal one, given by

$$S_i^{+} = \Big\{\sum_{j=1}^{m}\Big(R_{ij} - R_j^{+}\Big)^2\Big\}^{0.5}, \qquad i = 1(1)n;$$

  and

$$S_i^{-} = \Big\{\sum_{j=1}^{m}\Big(R_{ij} - R_j^{-}\Big)^2\Big\}^{0.5}, \qquad i = 1(1)n.$$

- Estimate the relative closeness of i^{th} alternative from ideal solution, given by

$$d_i = \frac{S_i^{-}}{S_i^{+} + S_i^{-}}.$$

The alternative having maximum d_i is considered as best feasible candidate.

4.6 Algorithm to generate Initial Population

This section presents the stochastic matrix-based initial population generation technique in view to apply variants of GA.

5 Numerical Examples

A company has two origins A_1 and A_2 with production capability of $[20, 27]$ and $[30, 40]$ units of manufactured goods, respectively. These units are to be transported to three warehouses B_1, B_2 andB_3 with necessity of $[15, 22], [16, 23.5]$ and $[11, 19.5]$ units, respectively. The conveyance capacity are $[20, 35]$ and $[31, 42]$.The coefficients for transportation cost, risk, weight, distance, and product impairment between companies to warehouses are given below:

Table 1: Transportation Cost for MOISTP

Warehouses	1		2		3	
Origins \Conveyances	1	2	1	2	1	2
1	[60,80]	[65,80]	[45,65]	[110,145]	[115,135]	[105,125]
2	[75,90]	[115,130]	[120,140]	[55,75]	[60,90]	[130,160]

Table 2: Risk Management for MOISTP

Warehouses	1		2		3	
Origins \Conveyances	1	2	1	2	1	2
1	[2,6]	[3,8]	[7,9]	[4,6]	[6,10]	[1,3]
2	[1,8]	[2,4]	[6,9]	[5,8]	[5,10]	[2,6]

Algorithm 1 Initial population procedure

Require: supply, demand, conveyance capacity
Ensure: Initial Population
  begin
  for $i = 1 : Population\ size$ **do**
    $o \leftarrow\ generate\ random\ numbers\ in\ the\ interval\ of\ supply\ and\ get\ sum\ S_1$;
    $D \leftarrow generate\ random\ numbers\ in\ the\ interval\ of\ demand\ having\ sum\ S_1$;
    $E \leftarrow generate\ random\ numbers\ in\ the\ interval\ of\ conveyance\ capacity\ having\ sum\ S_1$;

    $o_{count} \leftarrow 1 : numberoforigins$;
    $D_{count} \leftarrow column\ number\ of\ destination\ corresponding\ to\ conveyance$
    while $E(k) > 0$ **do**
      $temp1 \leftarrow randi(length(o_{count})$;
      if $(length(o) = length(o_{count}))$ **then**
        $row \leftarrow temp1$;
        $rowm = temp1$;
      else
        $row \leftarrow o_{count}(temp1)$;
        $rowm \leftarrow temp1$;
      end if
      $temp2 = randsample(D_{count}$,1);
      $Zn \leftarrow which\ destination\ temp2\ belongs$;
      $val \leftarrow min(E(k), o(rowm), D(Zn))$;
      $A(row, temp2) \leftarrow val$;
      $o(rowm) \leftarrow o(rowm) - val$;
      $D(temp2) \leftarrow D(temp2) - val$;
      $E(k) \leftarrow E(k) - val$;
      if $o(rowm) = 0$ **then**
        $o_{count}(rowm) = \phi$;
        $o_{rowm} = \phi$;
      end if
      if $D(Zn) = 0$ **then**
        $D_{count}(Zn) \leftarrow \phi$;
        $D(Zn) \leftarrow \phi$
      end if
    end while
  end for
  end

Table 3: Weight for MOISTP

Warehouses	1		2		3	
Origins \Conveyances	1	2	1	2	1	2
1	[16,30]	[12,30]	[65,85]	[40,60]	[70,95]	[15,45]
2	[20,25)]	[60,70]	[30,45]	[50,55]	[65,75]	[20,40]

Table 4: Distance for MOISTP

Warehouses	1		2		3	
Origins \Conveyances	1	2	1	2	1	2
1	[26,32]	[15,25]	[20,35]	[44,58]	[25,40]	[60,80]
2	[18,22]	[40,50]	[60,78]	[62,78]	[35,45]	[70,80]

Table 5: Product Deterioration for MOISTP

Warehouses	1		2		3	
Origins \Conveyances	1	2	1	2	1	2
1	[5,15]	[22,32]	[11,14]	[6,12]	[8,14]	[13,25]
2	[4,18]	[31,49]	[31,36]	[16,24]	[9,19]	[15,27]

Mathematical Model:

$$minZ_q(x) = \sum_{i=1}^{2}\sum_{j=1}^{3}\sum_{k=1}^{2} C_{ijk}^{q} x_{ijk}, q = 1(1)5, \quad (5.1)$$

subject to the constraints :

$$\sum_{j=1}^{3}\sum_{k=1}^{2} x_{1jk} \in [20, 27], \sum_{j=1}^{3}\sum_{k=1}^{2} x_{2jk} \in [30, 40], \quad (5.2)$$

$$\sum_{k=1}^{2}\sum_{i=1}^{2} x_{i1k} \in [15, 22], \sum_{k=1}^{2}\sum_{i=1}^{2} x_{i2k} \in [16, 23.5], \sum_{k=1}^{2}\sum_{i=1}^{2} x_{i3k} \in [11, 19.5], \quad (5.3)$$

$$\sum_{i=1}^{2}\sum_{j=1}^{3} x_{ij1} \in [20, 35], \sum_{i=1}^{2}\sum_{j=1}^{3} x_{ij2} \in [31, 42]. \quad (5.4)$$

First, MOISTP converted into MOSTP by risk attitude parameter taking $\epsilon = -.5$,$\epsilon = 0$ and $\epsilon = .5$ and then hybrid GA, NSGA II and NSGA III is employed. Parameters considered for NSGA III are as follows: population size = 50; crossover probability = 0.3; mutation probability = 0.03; number of division =6. These parameters are utilized for different values of ϵ. Moreover, some of the points from obtained Pareto-optimal set by NSGA III are shown in Table 6 to Table 8 for different choices of ϵ. Similarly, Table 9 to Table 11 consist of some of the points of the Pareto-optimal set obtained by NSGA II. Parameters considered for NSGA II are as follows: population size = 200; crossover probability = 0.3, mutation probability = 0.03.

Table 6: Pareto-optimal solution at $\epsilon = -.5$ by NSGA III

4172.758	193.3146	1885.169	1932.299	716.9919
4201.523	193.5056	1882.728	1932.274	724.6673
4442.229	166.2549	1905.337	2249.48	517.3752
4054.803	173.5517	1705.036	2244.082	612.4787
3995.034	166.7113	1710.493	2248.447	592.9819
3175.328	186.6915	1969.818	1933.165	450.8422
3866.659	153.3276	1810.107	2236.063	560.4185
4185.991	192.1287	1712.808	2173.63	642.343
4077.41	183.3319	1803.902	1953.354	537.2669
4274.427	222.6362	1640.626	1921.71	1014.064

Table 7: Pareto-optimal solution at $\epsilon = 0$ by NSGA III

3741.331	280.4838	2254.871	2190.944	667.4132
4216.965	276.7679	2143.697	2334.708	749.4979
4770.457	247.8016	2062.842	2012.135	622.5686
4210.085	284.9304	2188.162	2447.478	771.3898
4307.984	276.0567	2122.422	2362.22	765.206
4798.786	249.7939	2165.198	1991.951	666.3862
4474.017	276.2229	2245.231	2200.89	700.2395
4483.257	262.0955	2005.822	2179.404	761.5532
4345.934	334.8889	3241.572	1969.409	1040.836
3802.972	286.3489	2289.775	2245.34	695.2045

Table 8: Pareto-optimal solution at $\epsilon = .5$ by NSGA III

4707.517	383.0292	2586.8	2548.494	989.2785
4729.507	366.504	2489.61	2392.891	952.4577
6688.927	306.2654	2962.804	3245.542	1843.295
6688.927	306.2654	2962.804	3245.542	1843.295
5292.841	351.2026	2772.17	2494.751	1105.364
5152.04	339.7759	2097.729	2957.522	1173.897
5144.641	339.1754	2062.285	2981.842	1177.498
5190.602	350.7773	2232.7	2925.168	1160.369
5257.858	362.481	2377.226	2908.498	1154.449
5257.858	362.481	2377.226	2908.498	1154.449

Table 9: Pareto-optimal solution at $\epsilon = -.5$ by NSGA II

3210.058	165.781	1691.079	1726.579	500.972
4557.464	166.9588	1542.638	2005.658	619.4462
4579.574	154.5882	1574.225	2144.372	471.155
4557.167	154.6114	1559.481	2166.726	479.8447
4615.17	173.0231	1580.918	2051.824	603.4722
4224.08	194.7433	1680.181	2165.673	736.091
4140.266	165.2829	1776.215	1788.19	499.2957
4367.773	160.8408	1435.243	2408.179	620.3301
4089.581	174.7624	1578.85	2228.131	652.6932
4553.748	181.9624	1581.966	1918.24	735.3843
4870.144	201.1925	1414.927	2040.922	962.7044

Table 10: Pareto-optimal solution at $\epsilon = 0$ by NSGA II

3702.346	349.119	2684.437	1962.68	984.2696
4944.534	255.8408	1822.061	2096.504	816.3701
4555.749	242.9742	2150.361	1989.739	731.6226
4552.808	301.9633	1996.148	2124.38	1054.116
4469.105	242.7993	2106.79	2233.09	840.5512
4235.533	257.8503	2121.102	1961.329	870.4137
5065.888	237.6549	1906.455	2292.01	1027.888
5001.111	230.1068	1939.659	2146.962	688.1709
4878.852	280.1358	2146.462	2003.282	745.4483
4904.865	254.2774	1893.723	2467.425	880.2935

Table 11: Pareto-optimal solution at $\epsilon = .5$ by NSGA II

7015.539	309.0348	2771.038	3666.629	1883.84
6182.79	349.0385	2649.77	3035.058	1368.512
5171.132	359.2994	2968.285	2732.415	1289.409
5053.666	387.7185	2899.414	2573.162	1205.137
5607.438	383.3397	2353.092	3045.982	1367.67
5380.142	391.6898	2340.473	2882.596	1349.456
5096.708	367.993	3063.458	2626.239	1259.042
5615.726	403.9508	2506.539	2512.98	1154.852
5045.337	408.5937	2623.344	2451.766	1128.534
5798.41	362.784	2840.007	2862.9	1294.502

Discussion

After getting the Pareto-optimal set by NSGA III and NSGA II, we applied the topsis method to get the best feasible solution. To apply topsis, we give a weight of .2 to each objective. On the other side, the problem is also solved by fuzzy programming techniques using a linear membership function. Hybrid GA is employed to solved the problem using ALs $(.95, .92, .85, .89.73)$ and Sp $(-5, -10, -12, -8, -9)$. In solving the problem by all these approaches, we found that the number of Pareto-optimal solutions obtained by NSGA III is greater than NSGA II, even if we take less population size in NSGA III. Also, for $\epsilon = 0$ and $\epsilon = 0.5$, the best feasible solution obtained by NSGA III and the topsis method is better than in the three objectives from FPT. Furthermore, the best feasible solution obtained by NSGA III dominates the best feasible solution obtained by NSGA II for $\epsilon = 0.5$.

Table 13: Comparison table for MOISTP at $\epsilon = 0$

Method	Objective 1	Objective 2	Objective 3	Objective 4	Objective 5
FPT	4364.2	249.7917	1811.7	2293.1	895.3273
Hybrid GA	3612	248.7	2294.1	2215	937.2
NSGA II + Topsis	4555.749	242.9742	2150.361	1989.739	731.6226
NSGA III +Topsis	3741.331	280.4838	2254.871	2190.944	667.4132

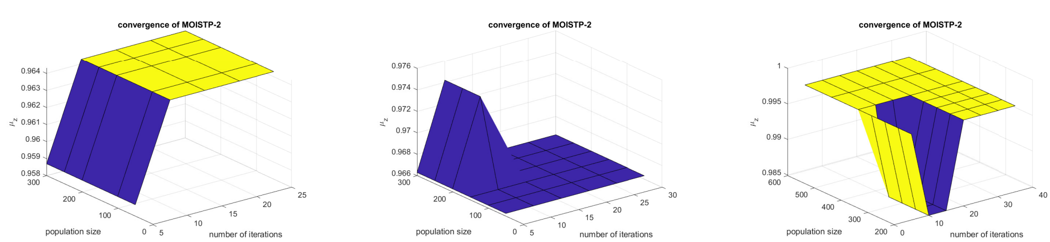
Table 14: Comparison table for MOISTP at $\epsilon = 0.5$

Method	Objective 1	Objective 2	Objective 3	Objective 4	Objective 5
FPT	4909.2	332.743	2245.1	2689.2	1171.5
Hybrid GA	5757.9	408.4	3170.4	2414.9	1870.2
NSGA II +Topsis	4861.786	403.9281	2600.213	2524.877	1163.998
NSGA III +Topsis	4729.507	366.504	2489.61	2392.891	952.4577

Table 12: Comparison Table at $\epsilon = -0.5$

Method	Objective 1	Objective 2	Objective 3	Objective 4	Objective 5
FPT	5092.5	127.9452	1483.5	1999.4	361.8776
Hybrid GA	4795.9	142.3	1162	2126.8	736.9
NSGA II +Topsis	4870.144	201.1925	1414.927	2040.922	962.7044
NSGA III +Topsis	3785.606	177.8997	2029.913	1670.469	391.5422

Figure 1a represents the convergence rate for the MOISTP at $\epsilon = -.5$. This graph is drawn among population size, iterations and the values of max $W = \prod_{i=1}^{5} Z_i$. Similarly, Figure 1b and 1c shows the convergence rate for the MOISTP at $\epsilon = 0$ and $\epsilon = 0.5$ respectively.



(a) Convergence Curve of MOISTP at $\epsilon = -0.5$

(b) Convergence Curve of MOISTP at $\epsilon = 0$

(c) Convergence Curve of MOISTP at $\epsilon = 0.5$

Figure 1: Convergence curve of MOISTP for different values of ϵ

6 Conclusion

This paper provides an algorithm for generating an initial feasible solution to the MOISTP in view of applying the genetic algorithm and its variants. Moreover, the topsis method is incorporated with NSGA II and NSGA III to get solutions that satisfy DM's preference/ weightage given to objectives. Finally, results are compared with hybrid GA and FPT, which shows the superiority of NSGA III over other adopted algorithms.

Compliance with ethical standards

Conflict of interest

The authors declare that they have no conflict of interest.

Ethical approval

This article does not contain any studies with human participants or animals performed by any of the authors.

Informed Consent

Shubha Agnihotri and Jayesh M. Dhodiya agree to submit this version and state that no part of the paper has been previously published or submitted. We thank you for taking the time to consider our work, and we eagerly await the reviewers' feedback as soon as possible.

References

[1] A. Baidya and U. Bera, An interval valued solid transportation problem with budget constraint in different interval approaches, *J. Transp. Secur.* **7**(2), 147–155 (2014).

[2] A. Baidya, U. Bera and M. Maiti, Solution of multi-item interval valued solid transportation problem with safety measure using different methods, *Opsearch*, **51**(1), 1–22 (2014).

[3] R. H. Bhesdadiya, I. N. Trivedi, P. Jangir, N. Jangir and A. Kumar, An NSGA-III algorithm for solving multi-objective economic/environmental dispatch problem, *Cogent Eng.* **3**(1), 1269383 (2016).

[4] G. Dantzig, Linear programming and extensions, *princeton universitypress* princeton, (1963).

[5] I. Das and J. E. Dennis, Normal-boundary intersection: A new method for generating the pareto surface in nonlinear multicriteria optimization problems. *SIAM J. Optim.* **8**(3), 631–657 (1998).

[6] K. Deb and H. Jain, An evolutionary many-objective optimization algorithm using referencepoint- based nondominated sorting approach, part i: solving problems with box constraints. *IEEE Trans. Evol. Comput.* **18**(4), 577–601 (2013).

[7] K. Deb, A. Pratap, S. Agarwal and T. Meyarivan, A fast and elitist multiobjective genetic algorithm: Nsga-ii, *IEEE Trans. Evol. Comput.* **6**(2), 182–197 (2002).

[8] C. M. Fonseca and P. J. Fleming, Genetic algorithms for multiobjective optimization: FormulationDiscussion and generalization, *In Icga.* **93**, 416–423, Citeseer (1993).

[9] P. Hajela, E. Lee and C.-Y. Lin, Genetic algorithms in structural topology optimization, *In Topology design of structures.* 117–133, Springer (1993).

[10] X. He, S. Dong and N. Zhao, Research on rush order insertion rescheduling problem under hybrid flow shop based on NSGA-III, *Int. J. Prod. Res.* **58**(4), 1161–1177 (2020).

[11] F. L. Hitchcock, The distribution of a product from several sources to numerous localities, *J Math Phys.* **20**(1-4), 224–230 (1941).

[12] K. Ji, W. Chen, X. Wu, H. Pang, J. Hu, S. Liu, F. Cheng and G. Tang, High frequency stability constraints based mmc controller design applying nsga-iii algorithm, *CSEE J. Power Energy Syst.* (2021).

[13] F. Jimenez, et al. Interval multiobjective solid transportation problem via genetic algorithms (1996).

[14] S. Kannan, S. Baskar, J. D. McCalley and P. Murugan, Application of NSGA-II algorithm to generation expansion planning, *IEEE Trans. Power Syst.* **24**(1), 454–461 (2008).

[15] A. Mart´ınez-Vargas, J. Dom´ınguez-Guerrero, ´A. G. Andrade, R. Sep´ulveda and O. Montiel-Ross, Application of NSGA-II algorithm to the spectrum assignment problem in spectrum sharing networks, *Appl. Soft Comput.* **39**, 188–198 (2016).

[16] A. Nagarajan and K. Jeyaraman, Solution of chance constrained programming problem for multi-objective interval solid transportation problem under stochastic environment using fuzzy approach, *Int. J. Comput. Appl.* **10**(9), 19–29 (2010).

[17] A. Nagarajan, K. Jeyaraman and S. K. Prabha, Multiobjective solid transportation problem with interval cost in source and demand parameters, *Int. j. comput. organization trends.* **8**(1), 33–41 (2014).

[18] J. D. Schaffer, Some experiments in machine learning using vector evaluated genetic algorithms, *Technical report.* Vanderbilt Univ., Nashville, TN (USA) (1985).

[19] E. Shell, Distribution of a product by several properties, directorate of management analysis, *In Proceedings of the second symposium in linear programming.* **2**, 615–642 (1955).

[20] H. Soyel, U. Tekguc and H. Demirel, Application of nsga-ii to feature selection for facial expression recognition, *Comput. Electr. Eng.* **37**(6), 1232–1240 (2011).

[21] N. Srinivas and K. Deb, Muiltiobjective optimization using nondominated sorting in genetic algorithms, *Evol Comput.* **2**(3), 221–248 (1994).

[22] S. Wang, D. Zhao, J. Yuan, H. Li and Y. Gao, Application of nsga-ii algorithm for fault diagnosis in power system. *Electr. Power Syst. Res.* **175**, 105893 (2019a).

[23] Y. Wang, C. Chen, Y. Tao, Z. Wen, B. Chen and H. Zhang, A many-objective optimization of industrial environmental management using nsga-iii: A case of china's iron and steel industry. *Appl. Energy.* **242**, 46–56 (2019b).

[24] P. Wangsom, K. Lavangnananda and P. Bouvry, The application of nondominated sorting genetic algorithm (nsga-iii) for scientific-workflow scheduling on cloud, *In The 8th Multidisciplinary International Conference on Scheduling: Theory and Applications (MISTA 2017).* 269–287, (2017).

[25] Jayesh M Dhodiya and Anita Ravi Tailor, Genetic algorithm based hybrid approach to solve fuzzy multi-objective assignment problem using exponential membership function, *SpringerPlus.* **5**(1), 1-29,(2016).

[26] Jeffrey Horn, Nicholas Nafpliotis and David E Goldberg, A niched Pareto genetic algorithm for multiobjective optimization, *Proceedings of the first IEEE conference on evolutionary computation. IEEE world congress on computational intelligence.* 82-87, Ieee, (1994).

Author information

Shubha Agnihotri and Jayesh M. Dhodiya, Department of Mathematics and Humanities, Sardar Vallabhbhai National Institute of Technology, Surat, Gujarat 395007, India.
E-mail: `shubha2371992@gmail.com, jmd@amhd.svnit.ac.in`