Yugoslav Journal of Operations Research
35 (2025), Number 1, 135–162
DOI: https://doi.org/10.2298/YJOR230218015T

Research Article

ASPIRATION LEVEL-BASED NON-DOMINATED SORTING GENETIC ALGORITHM II & III TO SOLVE FUZZY MULTI-OBJECTIVE SHORTEST PATH PROBLEM

Aniket S. TODKAR*
Department of Mathematics, Sardar Vallabhbhai National Institute of Technology, Surat, India
anikettodkar1001@gmail.com, ORCID: 0000-0003-2481-1898

Jayesh M. DHODIYA
Department of Mathematics, Sardar Vallabhbhai National Institute of Technology, Surat, India
jdhodiya2002@yahoo.com, ORCID: 0000-0001-9675-412X

Received: February 2023 / Accepted: April 2024

Abstract: The present article provides aspiration level-based non-dominated sorting genetic algorithm-II and III techniques for solving a fuzzy multi-objective shortest path problem utilizing an exponential membership function with a possibility distribution. Furthermore, in this study, using α-level sets, fuzzy judgement was categorized for the decision maker to simultaneously optimize fuzzy objective functions scenarios as optimistic, most likely, and pessimistic. A numerical illustration is presented together with a data set to demonstrate the use of the suggested techniques. A comparison is performed between the suggested methodology and several other approaches. The sensitivity of the objective functions is also investigated using aspiration levels and shape parameters. The coverage is computed to assess the effectiveness of the proposed methods. This research concludes that the suggested approaches can manage fuzzy multi-objective shortest path problems competently and efficiently with a solid yield, allowing the decision maker to make a decision based on its aspiration level.

*Corresponding author

Keywords: Multi-objective shortest path problem, fuzzy membership function, α- level set, triangular fuzzy number, aspiration level.

MSC: 68W50, 90B06, 90B50, 90C29.

1. INTRODUCTION

The challenge of discovering the Shortest Path (SP) between two vertices in a network is one of graph theory's most significant and crucial combinatorial network optimization problems. It has several applications in routing, wireless networks, supply chain management, transportation, and communications. Thousands of people suffer every day to navigate their cities' sources and destinations. Many software applications tackle this problem by representing the city as a graph and giving each arc on the graph a cost. Traditionally, it has been assumed that the arc traversal costs can be expressed as crisp numbers. The SP Problem (SPP) with crisp weights is known as conventional SPP. In the 1950s and 1960s, the examination of conventional SPP reviewed some robust computations made by Bellman [1], Dijkstra et al. [2], Floyd [3], and Dreyfus [4]. Edsger W. Dijkstra developed the Dijkstra algorithm [2] in 1956 for identifying the SPs between vertices in a graph, and it was published three years later. In most cases, the time required to complete a road journey is a cost. Google Maps, Waze, and TomTom are all navigation apps that work the same way. Another well-known component that can be gradually reduced using this technique is distance. The disadvantage of the Dijkstra algorithm is that it can not handle a negative weight edge graph. The Bellman-Ford algorithm [1] works on negative edges, but when dealing with negative cycle graphs, it does not produce the expected results. Floyd-Warshall algorithm [3] is used to compute the SP between each vertex, whereas the Dijkstra algorithm is used to compute the SP between a single vertex and each of the other vertices. Dijkstra algorithm has a significantly higher space overhead than the Floyd-Warshall algorithm.

Reducing travel time (or distance) is not the only crucial factor. Most individuals wish to reduce fuel expenses, travel on safer roads (risk), and simultaneously accomplish multiple objectives. As a result, in real-world situations, SPP relies on various factors; hence it is referred to as a multi-objective SPP (MOSPP). Hansen [5] was the first to describe the MOSPP in 1980. This work includes the primary label-setting algorithm for bi-objective SPPs. Martins [6] summed up the previous algorithm for MOSPP. Serafini [7] conveyed that the MOSPP is *NP*-complete. Later on, a few analysts contributed to the research of MOSPP [8, 9, 10, 11]. These works have significantly contributed to enhancing the importance of single/multi-objective SPP, which is crucial to the network theory. Subsequently, Sedeno-Noda and Colebrook [12] presented the bi-objective Dijkstra algorithm to reduce the hypothetical running time compared to the Dijkstra algorithm for the bi-objective SPP. The latter algorithm also exceeds the former algorithm in terms of computational studies. Various ways to deal with MOSPP could be multi-stage approaches utilizing inclination-based enhancement as in Pugliese et al. [13] or swarm intelligence graph-based algorithm as introduced by Ntakolia and Iakovidis [14]. Recently, De las Cases et al. [15] presented an extension of bi-objective Dijkstra algorithm, a new label-setting algorithm that is a multi-objective Dijkstra algorithm, to compute a minimum complete set of efficient pathways for MOSPP.

Various real-life instances demand us to deal with uncertain parameters, like strikes, traffic congestion, flood, poor visibility in the winter, uneven roads, road accidents, or bad human health travelling from one place to another. Consequently, the quantities time, cost, distance, and risk are imprecise. Fuzzy sets address these sorts of imprecision in data. As a result, the SPP has been converted into a Fuzzy SPP (FSPP) in which fuzzy numbers express objective weights. The FSPP was first examined by Dubois [16] in 1980. However, even if the shortest distance can be found, a corresponding SP cannot be found. Okada and Soper [17] invented an algorithm that uses multiple labelling to restrict the number of paths depending on a possibility level. Researchers like Mahdavi et al. [18] and Tajdin et al. [19] used a dynamic programming approach to develop a model and algorithm for determining the SP in a network with various types of fuzzy arc lengths. Likewise, Mukherjee [20] addressed FSPP by a methodology called the fuzzy programming technique. Ebrahimnejad et al. address FSPP by particle swarm optimization algorithm [21] and artificial bee colony algorithm [22]. Several authors [23, 24, 25] etc. have worked in the field of FSPP. Ebrahimnejad et al. solved SPP in various fuzzy environments like interval-valued fuzzy networks [26], interval-valued Pythagorean fuzzy environment [27], interval-valued triangular fuzzy network [28], and mixed interval-valued fuzzy environment [29]. Sori et al. [30] solved the robot's fuzzy constrained routing problem by an elite artificial bee colony algorithm. Ebrahimnejad [31] proposed a generalized Dijkstra algorithm to solve SPP with interval weights. Recently, Lin et al. [32] solved FSPP by genetic algorithm, and Di et al. [33] solved FSPP by ant colony optimization algorithm.

Fuzzy MOSPP (FMOSPP) is a MOSPP with at least one fuzzy parameter. There are very few methods available for solving FMOSPPs. Rani and Reddy [34] examined the FMOSPP, a bi-objective optimization problem with crisp and trapezoidal fuzzy values. It is used to demonstrate the techniques depending on the priority and type of the problem so that the DM can choose the most satisfactory or best solution. Some authors [35, 36, 37] contributed their work in the field of extended fuzzy MO problems. Recently, Bagheri et al. [38] solved FMOSPP dependent on the data envelopment analysis approach. They converted FMOSPP into a single objective FSPP that can be solved using existing FSPP methods. In literature, the FMOSPP was solved by converting it into a single objective to get a single solution. However, in this case, the DM does not have the choice to select other non-dominated solutions. Furthermore, suppose the network contains large numbers of vertices and edges. In that case, some existing methods become very complicated for computation (for example, fuzzy programming technique), and they take more time to solve this problem. Evolutionary genetic approaches are particularly effective in all these regard since they evolve toward better solutions by utilizing genetic operators based on natural genetic processes. The genetic algorithm-based hybrid approach gives us a single solution. The methods Non-dominated Sorting Genetic Algorithm (NSGA)-II & NSGA-III provide a Pareto frontier, i.e., all non-dominated solutions in the first front; thus, DM also receives solutions below its Aspiration Level (AL). In order to overcome these limitations of existing approaches, this article deals with modified solution techniques for FMOSPP. This work proposes AL-based NSGA-II and AL-based NSGA-III. These methods aim to find the Pareto frontier for FMOSPP, which satisfies the DM's AL so that the Decision Maker (DM) can choose the solution as per requirement.

This study aims to present an algorithmic approach for MOSPP in a fuzzy environment that is effective for practical problems. This article has contributed the following: (1) FMOSPP has been formulated and solved by considering the arc weights as TFN in the given network. (2) The initial population has been produced utilizing the new method. (3) A significant modification has been made to the selection operator to minimize the program's execution time. (4) AL-based NSGA-II and AL-based NSGA-III evolutionary techniques have been developed to overcome the limitations of Hybrid Genetic Algorithm (HGA), NSGA-II, and NSGA-III, respectively.

The remaining paper is organized as follows. Section 2 represents the mathematical model of FMOSPP. In Section 3, the preliminary concepts of the possibilistic programming approach, triangular possibilistic distribution, α-level set, Positive Ideal Solution (PIS) & Negative Ideal Solution (NIS), and Exponential Membership Function (EMF) are discussed. Section 4 deals with the formulation of the Multi-Objective (MO) 0-1 programming model. Section 5 describes solution methods of the auxiliary model, namely the proposed AL-based NSGA-II, and AL-based NSGA-III. Section 6 presents benchmark instances and computational complexity of proposed methods. A numerical example, its solutions, and results & discussion are described in Section 7. Lastly, Section 8-11 represents the sensitivity analysis, comparison, performance measure, and conclusion, respectively.

2. MATHEMATICAL MODEL OF FMOSPP

Let, $G = (V, E)$ be a network, where V represents set of vertices and E represents set of arcs. Assume that, the network contains s vertices & r arcs. Consider the starting point is vertex 1, and the ending point is vertex s and our aim is to discover the SP between these two vertices. A unit of flow enters from outside the network G at vertex 1 and exits at vertex s. In any arc, only one unit of flow can be present at a time, so the decision variable should expect binary qualities (0 or 1) in particular. There exists one restriction that covers flow preservation at every vertex: total input flow equals to total output flow for each vertex $u \in V$. Define, x_{uv} amount of flow in arc (u, v) $\forall$ feasible u and v, $\tilde{c_{uv}}$ cost per unit of flow in the arc (u, v) $\forall$ feasible u and v in the form of TFN. Consider the four objective functions: time $\tilde{t_{uv}}$, cost $\tilde{c_{uv}}$, distance $\tilde{d_{uv}}$, and risk $\tilde{r_{uv}}$, which are in form of TFN. Thus, the mathematical formulation of FMOSPP [39] is given by,

Formulation of objective function:

$$\mathit{min} \sum_{u=1}^{s} \sum_{v=1, v \neq u}^{s} \tilde{t_{uv}} x_{uv}, \ \mathit{min} \sum_{u=1}^{s} \sum_{v=1, v \neq u}^{s} \tilde{c_{uv}} x_{uv},$$

$$\mathit{min} \sum_{u=1}^{s} \sum_{v=1, v \neq u}^{s} \tilde{d_{uv}} x_{uv}, \ \mathit{min} \sum_{u=1}^{s} \sum_{v=1, v \neq u}^{s} \tilde{r_{uv}} x_{uv}$$

Model constraints:

$$\sum_{v=1}^{r} x_{1v} - \sum_{k=1}^{s} x_{k1} = 1, \tag{1}$$

$$\sum_{v=1}^{r} x_{uv} - \sum_{k=1}^{s} x_{ku} = 0, \quad u \neq 1, s\ , \tag{2}$$

$$\sum_{v=1}^{r} x_{sv} - \sum_{k=1}^{s} x_{ks} = -1, \tag{3}$$

$$x_{uv} = 0\ or\ 1,\ u, v = 1, 2, ..., s. \tag{4}$$

$x_{uv} = 0\ or\ 1$, indices that arc (u,v) is in path or not respectively.
$\tilde{t_{uv}} = (t^o_{uv}, t^m_{uv}, t^p_{uv})$, $\tilde{c_{uv}} = (c^o_{uv}, c^m_{uv}, c^p_{uv})$, $\tilde{d_{uv}} = (d^o_{uv}, d^m_{uv}, d^p_{uv})$, $\tilde{r_{uv}} = (r^o_{uv}, r^m_{uv}, r^p_{uv})$, denotes TFN for time, cost, distance, risk parameter respectively.

Decision Problem:
Model 1:
The FMOSPP is now formulated as follows:

$$min(\tilde{z_1}, \tilde{z_2}, \tilde{z_3}, \tilde{z_4}) = min\Big(\sum_{u=1}^{s} \sum_{v=1, v\neq u}^{s} \tilde{t_{uv}} x_{uv}, \sum_{u=1}^{s} \sum_{v=1, v\neq u}^{s} \tilde{c_{uv}} x_{uv},$$
$$\sum_{u=1}^{s} \sum_{v=1, v\neq u}^{s} \tilde{d_{uv}} x_{uv}, \sum_{u=1}^{s} \sum_{v=1, v\neq u}^{s} \tilde{r_{uv}} x_{uv}\Big)$$

subject to the constraints (1)-(4).

3. PRELIMINARIES

3.1. Possibilistic Programming Approach

In most cases, collecting data on real-world situations involves some level of risk. Because of their nature, many forms of data cannot be specified and are hence represented by fuzzy numbers. A possibility distribution is used to model these sorts of fuzzy numbers [40]. The probabilistic distribution has been utilized for solving fuzzy advancement models with uncertain coefficients in the objective function with a wide variety of key applications. Using the possibilistic technique, the FMOSPP model was turned into an auxiliary crisp MO Optimization (MOO) model [41].

3.2. Triangular Possibilistic Distribution (TPD)

Since the uncertain parameters are not defined precisely, the TPD is usually utilized because of its effortlessness and computational viability in acquiring information. In sensible conditions, a DM can build the TPD by utilizing the most optimistic value (o) (possibility degree = 0), most likely value (m) (possibility degree = 1), and the most pessimistic value (p) (possibility degree = 0) respectively which is generally denoted by (c^o_i), (c^m_i), and (c_i^p). From Figure 1, at three positions $(c_i^m, 1)$, $(c_i^o, 0)$ and $(c_i^p, 0)$ defined an objective function cost which is minimized by moving the three places of TPD to the left since vertical directions of the focuses are fixed by 0 or 1. Consequently, just the three horizontal coordinates are assumed.

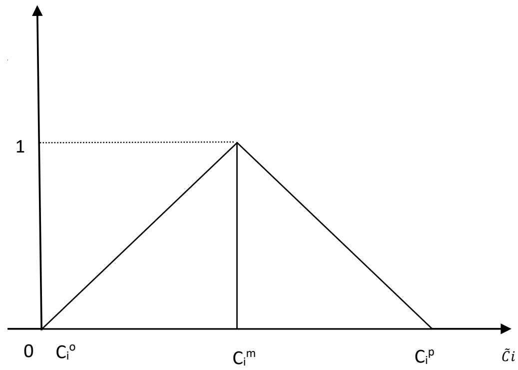


Figure 1: TPD of $\tilde{c}_i$

3.3. α-level Set

Zadeh [42] provided a α-level set which is the fundamental hypothesis for establishing a relationship between traditional and fuzzy set theories. The α-level, also known as the confidence level, represents the DM's assurance regarding his fuzzy judgement. The lowest α-value indicates a significant amount of pessimism and uncertainty by providing an interval judgement with huge savings. The greatest α-value results in a lower but more optimistic judgement, with the lower and upper bounds having a higher degree of membership in the original fuzzy sets. Some researchers, Lai and Hwang [43]; Tailor and Dhodiya [44]; Rekh and Dhodiya [45] employed this α-level concept for addressing fuzzy optimization problems. Similarly, this idea is utilized in the current article to decide the DM's confidence in his fuzzy judgement.

3.4. Positive and Negative Ideal Solution

The objective function's minimum value is known as PIS, while its maximum value is known as NIS. For every objective function, both values are utilized to derive the value of the membership function.

3.5. Exponential Membership Function

The EMF $\mu_{z_k}(x)$ can be used to normalize data relating to the given problem. If the number z_k^{NIS} and z_k^{PIS} stands for NIS and PIS respectively for objective z_k , then the $\mu_{z_k}(x)$ is represented as below,

$$\mu_{z_k}(x) = \begin{cases} 1, & if\ z_k \le z_k^{PIS}, \\ \frac{e^{-s_k \psi_k(x)} - e^{-s_k}}{1-e^{-s_k}}, & if\ z_k^{PIS} < z_k < z_k^{NIS}, \\ 0, & if\ z_k \ge z_k^{NIS}. \end{cases} \tag{5}$$

where, $\psi_k(x) = \frac{z_k - z_k^{PIS}}{z_k^{NIS} - z_k^{PIS}}$, $0 \leq \mu_{z_k}(x) \leq 1$ and $s_k \neq 0$, the DM's shape parameter. Furthermore, the membership function will be convex and concave as appropriate for $s_k < 0 \; and \; s_k > 0$ in $[z_k^{PIS}, z_k^{NIS}]$.

4. FORMULATION OF MO 0-1 PROGRAMMING MODEL

The TPD strategy handles fuzzy objectives to formulate the auxiliary crisp MOO model of Model 1. The objective function of distance is represented as,

$$min \; \tilde{z_3} = min(z_3^o, z_3^m, z_3^p) = \sum_{u=1}^{s} \sum_{v=1, v \neq u}^{s} \tilde{d_{uv}} x_{uv}$$
$$= min(\sum_{u=1}^{s} \sum_{v=1, v \neq u}^{s} d_{uv}^o x_{uv}, \sum_{u=1}^{s} \sum_{v=1, v \neq u}^{s} d_{uv}^m x_{uv}, \sum_{u=1}^{s} \sum_{v=1, v \neq u}^{s} d_{uv}^p x_{uv}), \tag{6}$$

where $\tilde{d_{uv}} = (d_{uv}^o, d_{uv}^m, d_{uv}^p)$, this can be also written as,

$$(min \; z_{31}, min \; z_{32}, min \; z_{33}) = min\big(\sum_{u=1}^{s} \sum_{v=1, v \neq u}^{s} d_{uv}^o x_{uv}, \sum_{u=1}^{s} \sum_{v=1, v \neq u}^{s} d_{uv}^m x_{uv},$$
$$\sum_{u=1}^{s} \sum_{v=1, v \neq u}^{s} d_{uv}^p x_{uv}\big), \tag{7}$$

The equations (6) and (7) represented the optimistic, most likely, and pessimistic scenario.

Utilizing the α-level set scenario, each $\tilde{d_{uv}}$ can be written as $(\tilde{d_{uv}})_\alpha = ((d_{uv})_\alpha^o, (d_{uv})_\alpha^m,$ $(d_{uv})_\alpha^p)$, where $(d_{uv})_\alpha^o = (d_{uv})^o + \alpha((d_{uv})^m - (d_{uv})^o)$, $(d_{uv})_\alpha^m = (d_{uv})^m$, $(d_{uv})_\alpha^p = (d_{uv})^p - \alpha((d_{uv})^p - (d_{uv})^m)$. Hence equation (7) becomes:

$$(min \; z_{31}, min \; z_{32}, min \; z_{33}) = min\big(\sum_{u=1}^{s} \sum_{v=1, v \neq u}^{s} (d_{uv})_\alpha^o x_{uv}, \sum_{u=1}^{s} \sum_{v=1, v \neq u}^{s} (d_{uv})_\alpha^m x_{uv},$$
$$\sum_{u=1}^{s} \sum_{v=1, v \neq u}^{s} (d_{uv})_\alpha^p x_{uv}\big) \tag{8}$$

Similarly, the MOO Problem (MOOP) model of time, cost, and risk objective functions are as follows:

$$(min\ z_{11}, min\ z_{12}, min\ z_{13}) = min\Big(\sum_{u=1}^{s}\sum_{v=1,v\neq u}^{s}(t_{uv})^{o}_{\alpha}x_{uv}, \sum_{u=1}^{s}\sum_{v=1,v\neq u}^{s}(t_{uv})^{m}_{\alpha}x_{uv}, \sum_{u=1}^{s}\sum_{v=1,v\neq u}^{s}(t_{uv})^{p}_{\alpha}x_{uv}\Big), \quad (9)$$

$$(min\ z_{21}, min\ z_{22}, min\ z_{23}) = min\Big(\sum_{u=1}^{s}\sum_{v=1,v\neq u}^{s}(c_{uv})^{o}_{\alpha}x_{uv}, \sum_{u=1}^{s}\sum_{v=1,v\neq u}^{s}(c_{uv})^{m}_{\alpha}x_{uv}, \sum_{u=1}^{s}\sum_{v=1,v\neq u}^{s}(c_{uv})^{p}_{\alpha}x_{uv}\Big), \quad (10)$$

$$(min\ z_{41}, min\ z_{42}, min\ z_{43}) = min\Big(\sum_{u=1}^{s}\sum_{v=1,v\neq u}^{s}(r_{uv})^{o}_{\alpha}x_{uv}, \sum_{u=1}^{s}\sum_{v=1,v\neq u}^{s}(r_{uv})^{m}_{\alpha}x_{uv}, \sum_{u=1}^{s}\sum_{v=1,v\neq u}^{s}(r_{uv})^{p}_{\alpha}x_{uv}\Big). \quad (11)$$

Auxiliary MO 0-1 programming model: To formulate crisp MOSPP, known as auxiliary MO 0-1 programming model, from FMOSPP by utilizing α-level set to obtain the optimistic, most likely, and pessimistic scenarios, which is represented by the following:

Model 2:

$$\begin{aligned}(min\ z_{11}, min\ z_{12}, min\ z_{13}, min\ z_{21}, min\ z_{22}, min\ z_{23},\\ min\ z_{31}, min\ z_{32}, min\ z_{33}, min\ z_{41}, min\ z_{42}, min\ z_{43}) =\\ min\Bigg(\sum_{u=1}^{s}\sum_{v=1,v\neq u}^{s}(t_{uv})^{o}_{\alpha}x_{uv}, \sum_{u=1}^{s}\sum_{v=1,v\neq u}^{s}(t_{uv})^{m}_{\alpha}x_{uv}, \sum_{u=1}^{s}\sum_{v=1,v\neq u}^{s}(t_{uv})^{p}_{\alpha}x_{uv},\\ \sum_{u=1}^{s}\sum_{v=1,v\neq u}^{s}(c_{uv})^{o}_{\alpha}x_{uv}, \sum_{u=1}^{s}\sum_{v=1,v\neq u}^{s}(c_{uv})^{m}_{\alpha}x_{uv}, \sum_{u=1}^{s}\sum_{v=1,v\neq u}^{s}(c_{uv})^{p}_{\alpha}x_{uv},\\ \sum_{u=1}^{s}\sum_{v=1,v\neq u}^{s}(d_{uv})^{o}_{\alpha}x_{uv}, \sum_{u=1}^{s}\sum_{v=1,v\neq u}^{s}(d_{uv})^{m}_{\alpha}x_{uv}, \sum_{u=1}^{s}\sum_{v=1,v\neq u}^{s}(d_{uv})^{p}_{\alpha}x_{uv},\\ \sum_{u=1}^{s}\sum_{v=1,v\neq u}^{s}(r_{uv})^{o}_{\alpha}x_{uv}, \sum_{u=1}^{s}\sum_{v=1,v\neq u}^{s}(r_{uv})^{m}_{\alpha}x_{uv}, \sum_{u=1}^{s}\sum_{v=1,v\neq u}^{s}(r_{uv})^{p}_{\alpha}x_{uv}\Bigg)\end{aligned} \quad (12)$$

subject to the constraints (1)-(4).

5. SOLUTION METHODS FOR AUXILIARY MODEL

This section describes the proposed AL-based NSGA-II and III, which have been implemented to solve an FMOSPP. These methods use a randomly generated population of feasible solutions that "evolve" generation by generation toward a better solution. As a result, the chromosome encoding and generation of the population are crucial for these methods. Generally, these methods can be applied to a wide variety of multi-objective optimization problems such as assignment problems, transportation problems, travelling salesman problems, etc. As some problems may require different chromosome encodings, initial populations, and genetic operators (selection, crossover, and mutation), these may change, but the procedure of methods will be the same.

5.0.1. Chromosome Encoding

Every chromosome specifies a path between the starting vertex and the ending vertex. It is a feasible solution that may or may not be optimal. The binary encoding has been utilized to create a chromosome for FMOSPP. Let x_{uv} denote the arc from vertex u to vertex v. Create a row vector that contains all the arcs from a given network orderwise. Consider one feasible path; if arc x_{uv} is present in the path, the entry is 1, otherwise 0.

Example: Consider the network in Figure 2, and $1 \rightarrow 2 \rightarrow 4 \rightarrow 5 \rightarrow 7 \rightarrow 8$ is one feasible path. Generate a row vector that contains all the arcs order-wise from Figure 2 i.e. $(x_{12}, x_{23}, x_{24}, x_{34}, x_{36}, x_{45}, x_{46}, x_{57}, x_{67}, x_{78})$. In the above feasible path, the arcs x_{12}, x_{24}, x_{45}, x_{57} and x_{78} are present, so it is replaced by 1 and others are replaced by 0. As a result, the chromosome becomes $[1\ 0\ 1\ 0\ 0\ 1\ 0\ 1\ 0\ 1]$.

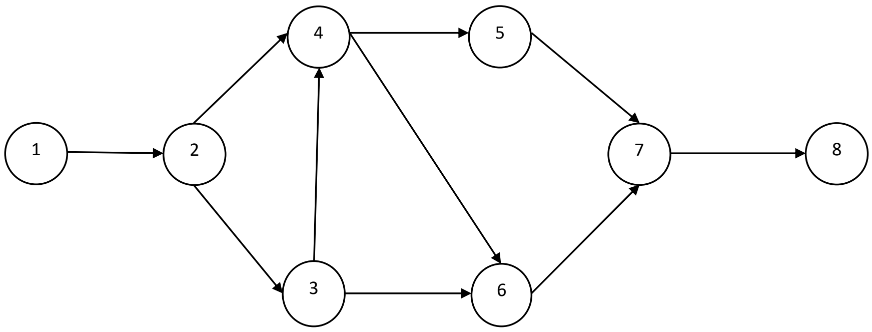


Figure 2: Network Diagram

5.0.2. Initial population

In this study, the initial population was randomly generated. For that, vectors of size $1 \times$ (number of arcs) containing entries 0 & 1 are generated using MATLAB $\mathcal{R}$ times ($\mathcal{R}$ is a predefined number that is initially defined for a total number of randomly generated vectors). Check whether the randomly generated vectors are feasible or not. The distinct feasible vectors are directly included in the population, and others are discarded. If the population size falls short, repeat the feasible vectors until the population size is satisfied.

5.1. AL-based NSGA-II

Srinivas and Deb [46] developed NSGA based on Goldberg's ideas of applying the non-dominated sorting concept in GAs [47]. The disadvantage of the NSGA method is that the σ_{share} parameter must be fixed when using the sharing function technique. The performance of an NSGA has previously been shown to be sensitive to the parameter σ_{share} [46]. In order to overcome this, Deb et al. [48] developed a new method called

NSGA-II. The NSGA-II approach provides a Pareto frontier. Therefore, DM also receives solutions below its AL. Thus, the authors modify the NSGA-II to fulfill DM's AL by adding a AL-based constraint $\mu(x) - \overline{\mu}(x) \geq 0$, where $\overline{\mu}(x)$ is given AL, so that DM may pick solutions that satisfy its AL, and this updated method is called AL-based NSGA-II.

For the sake of simplicity, few terms have been defined below in 5.1.1 and 5.1.2. The crowded distance assignment and procedure of the above method is described in 5.1.3 and 5.1.4.

5.1.1. Feasibility test

This test determines whether the solution to the given problem is feasible or not.

5.1.2. Updation of population

Suppose I be the population of size N and x_1 is any member of I. If x_1 satisfies the AL-based constraint, it is directly incorporated into updated I and denoted by $\overline{I}$. If the population size of $\overline{I}$ is not equal to N, then fulfil the remaining size by adding constraints satisfying x_i repeatedly.

5.1.3. Crowded distance assignment

A solution p wins a competition against another solution q if one of the below conditions is valid:

1. If a solution p has a superior rank.
2. If they have a similar rank, however, solution p has a superior crowding distance than solution q.

Crowded distance assignment procedure

1. Call $l = Cardinality(F)$ for the number of solutions in F . For every p in the set, initially allot crowding distance $d_p = 0$.
2. In worse order of F_r sort the set for every objective function $r = 1, 2 \cdots R$.
3. For $r = 1, 2 \cdots R$ assign a high distance to the border solutions, then for all remaining solutions from $q = 2$ to $l - 1$, assign

$$d_{(I_q^r)} = d_{(I_q^r)} + \frac{f_r^{(I_{q+1}^r)} - f_r^{(I_{q-1}^r)}}{f_r^{max} - f_r^{min}}. \quad (13)$$

The index I_q represents the solution index of the sorted list's q^{th} item.

5.1.4. AL-based NSGA-II procedure

A random initial population I_o of size N is generated initially. Update the population I_o to $\overline{I_o}$ using 5.1.2. The population $\overline{I_o}$ is then divided into separate non-domination fronts, and each individual of $\overline{I_o}$ is assigned a fitness level corresponding to their non-domination front. Following that, genetic operations like selection, crossover, and mutation generate the offspring population O_o from $\overline{I_o}$ of size N. Test the feasibilty of O_o and then update it to $\overline{O_o}$ using 5.1.2. Combine $\overline{I_o}$ and $\overline{O_o}$ to make the merging set M_o. At generation t, the parent population is $\overline{I_t}$, the offspring population is $\overline{O_t}$, and the merge set is $M_t (= \overline{I_t} U \overline{O_t})$

with a size of $2N$. With the aid of the selection operator, i.e., crowding comparison operator, the N best individuals are chosen for the next generation among these $2N$ individuals. This operator utilizes two criteria,

i) non-domination level/rank j_{rank}

ii) crowding distance $j_{distance}$.

In order to achieve this, sort M_t into distinct non-domination fronts. The fitness of individuals in M_t is equivalent to their non-domination front. So, F_j is the distinct non-domination front, and $j = 1, 2, ...,$ & so on. Any individual's non-domination front determines their j_{rank} in M_t. Assign a j_{rank} to each individual in M_t according to their fronts. Individuals with a smaller j_{rank} are favoured over those with higher j_{rank}. Thus front F1 with individuals of j_{rank} 1 is favoured over front F2 with individuals of j_{rank} 2 and so on for selection in next-generation I_{t+1}. Now, in I_{t+1}, add fronts $F_1, F_2, ...,$ & so on one by one until the size of I_{t+1} equals or exceeds N for the first time. Assume that F_k is the last front added in I_{t+1} and that all fronts after F_{k+1} are rejected. If the size of I_{t+1} is precisely N, then our $I_{t+1} = F_1 \cup F_2 \cup ... \cup F_k$. If the size of I_{t+1} exceeds, then pick solutions based on their crowding distance criteria (say $j_{distance}$) since j_{rank} is the same for all individuals in F_k. From the front F_k, $N - |F_1 \cup F_2 \cup \ldots \cup F_k|$ individuals are now required. If a and b are individuals in F_k and $a_{rank} = b_{rank}$, but $a_{distance} > b_{distance}$, then a is preferred over b in I_{t+1}. The individuals with the highest crowding distance are eventually chosen from F_k to fill the remaining individuals in I_{t+1}. Finally, for the following generation, the individuals of I_t are replaced with those of I_{t+1}. Repeat this procedure till it arrives at the stopping criteria.

This manuscript slightly modified the selection operator to minimize the time factor. When applying genetic operators to generate the new chromosomes, increase the size of the original population equal to the number of random vectors generated at the initial population (i.e., $\mathcal{R}$) by taking chromosomes repeatedly. Later on, for the next step of non-dominating sorting, the population first selects non-repeated chromosomes from the original and newly formed child populations. To fulfill the original population size, randomly select the non-repeating chromosomes again.

Algorithm: The following is the algorithm of AL-based NSGA-II to solve Model 2:

Algorithm 1 AL-based NSGA-II

Require: Objective function, Population size, AL ($\overline{\mu}$), Shape parameter
Ensure: $X, minZ_{uv}$

Read: Model 2
Generation = 0; Generate random initial population I_0
for $x \in I_0$ **do**
  Evaluate fitness values (f_1) of x
  Find PIS and NIS for every objective function of Model 2
  Find EMF $\mu(x)$
end for
Update I_0 to $\overline{I_0}$ using 5.1.2
while Termination condition not met **do**
  Generation $\leftarrow$ Generation+1;
  Set $f \leftarrow$ By applying selection procedure of NSGA-II on $\overline{I_0}$
  for $x_1, x_2 \in f$ **do**
    $x_1^{'}, x_2^{'} \leftarrow$ new offsprings by utilizing crossover on x_1 & x_2
    $I_0^{'} = \cup x_u^{'}$ where $x_u^{'}$ is new offsprings from above step
  end for
  for $y_1 \in f$ **do**
    $y_1^{'} \leftarrow$ new offsprings by utilizing bit mutation on y_1
    $I_0^{''} = \cup y_v^{'}$ where $y_v^{'}$ is new offsprings from above step
  end for
  $O_0 \leftarrow I_0^{'} \cup I_0^{''}$
  $O_0 \leftarrow$ Check the feasibility test for O_0
  Update O_0 to $\overline{O_0}$ using 5.1.2
  At generation t, $\overline{I_t} \leftarrow$ parent population and $\overline{O_t} \leftarrow$ offspring population. Merge set $M_t \leftarrow \overline{I_t} \cup \overline{O_t}$
  Do non-dominated sorting on M_t
  $I_{t+1} \leftarrow$ selection procedure of NSGA-II on M_t to select N best solution
end while

5.2. AL-based NSGA-III

Deb and Jain [49] developed an NSGA-III algorithm for sustaining population diversity that uses a reference-point strategy. NSGA-III generates the Pareto front. As a result, DM gets solutions that are lower than its AL. This study proposes the AL-based NSGA-III to prioritize the DM's AL by adding an AL-based constraint in the NSGA-III method.

Definition 1. *In objective space, a point specified by a DM and provides his/her ALs toward objective functions is a Reference Point (RP). The RP-based algorithms seek to provide a non-dominated solution set that is as close to an RP [50].*

NSGA-III begins with a randomly generated population of size N and a collection of broadly dispersed pre-specified M-dimensional RPs H on a unit hyper-plane with a normal vector of one's spanning the whole RM^+ area. The hyper-plane touches one point of every objective axis configured. The Das and Dennies [51] approach is used to locate H = $\binom{m+p+1}{p}$ RPs on the hyper-plane with $(p+1)$ points along every border. The population size N is selected to be the lowest multiple of four greater than H, with each RP having a chance of discovering one population member.

Consider at generation t, where the parent population is indicated by I_t with cardinality N. Update I_t to $\overline{I_t}$ using 5.1.2. The offspring population is denoted by O_t, which is generated from $\overline{I_t}$ using genetic operations like crossover and mutation. Check the feasibility test for O_t and update O_t to $\overline{O_t}$ using 5.1.2. Now, merge the parent population $\overline{I_t}$ and offspring population $\overline{O_t}$ to form M_t (i.e. $M_t = \overline{I_t} \cup \overline{O_t}$) of cardinality $2N$. The main goal of this method is to determine how to choose N members from M_t for the next generation. To get non-domination fronts, do Pareto-based non-dominating sorting on M_t, i.e., $F_1, F_2, ...,$ & so on. At this point, an empty population S_t is established. An individual from non-domination levels is added to S_t one by one, starting with F_1, until the size of S_t approaches or surpasses N for the first time. Assume the latest level added to S_t is F_l, and all fronts from the level $(l+1)^{th}$ onwards are rejected. The final level accepted is l^{th}, which is only partially accepted in certain instances. The members of the S_t/F_l population are added to the new population I_{t+1}, and the diversity maintenance operator selects the remaining individuals from F_l. Use the normalization operator to prepare the environment selection, keeping the RP and objective points in the same unit range. The zero vector is the ideal population point S_t after normalization and specified RPs lie on this normalized hyper-plane. The perpendicular distance of every individual in S_t from each RP line (connecting the RP with the origin) was computed. It was concluded that individuals are associated with RPs with the shortest perpendicular distance. A niche preservation process was utilized to select individuals from F_l. ρ_j is the niche count for the j^{th} RP and is defined as number of individuals associated with j^{th} RP from S_t/F_l. First, identified the minimum ρ_j value from the RPs set $J_{min} = \{j : argmin_j \rho_j\}$. If $|J_{min}| > 1$ then randomly choose $j^- \in J$.
The following two situations are then utilized:

- If, with the j^{th} RP, some individuals in F_l are associated then assume two cases:
  1. If $\rho_j = 0$, the one individual from F_l having minimum perpendicular distance from j^{th} reference line add up into I_{t+1}. After that count of ρ_j is increased by one.
  2. If $\rho_j > 0$, randomly choose one individual from F_l which is associated with j^{th} RP and add up into I_{t+1}. After that count of ρ_j is increased by one.
- If, with the j^{th} RP, no individuals in F_l are associated, then exclude the present RP for the current generation, for the moment J_{min} is recalculated, and j^- is reselected.

Repeated this procedure until the remaining individuals of I_{t+1} are filled up. In this manner, we keep evolving better solutions over generations till we arrive at stopping criteria.
Algorithm:
The algorithm for the AL-based NSGA-III to solve Model 2 is the same as AL-based NSGA-II; the only difference is that the selection procedure which occcures in steps 11 and 25.

Flow chart of the solution procedure of FMOSPP by proposed methods AL-based NSGA-II & III are given in Figure 3.

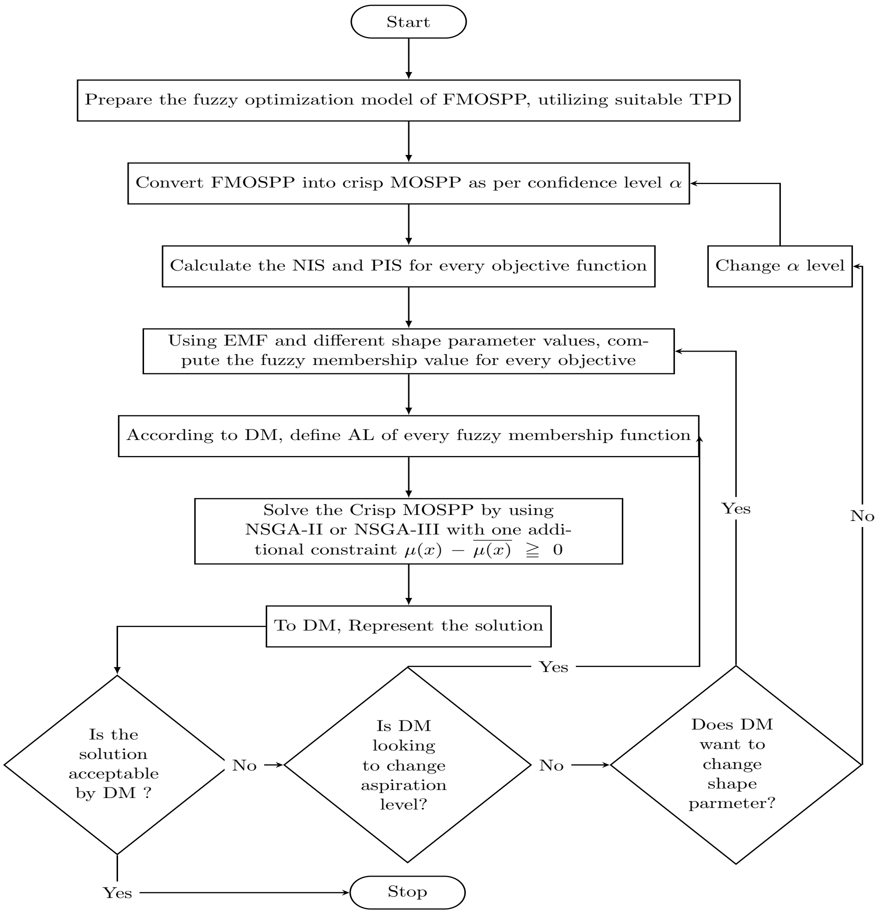
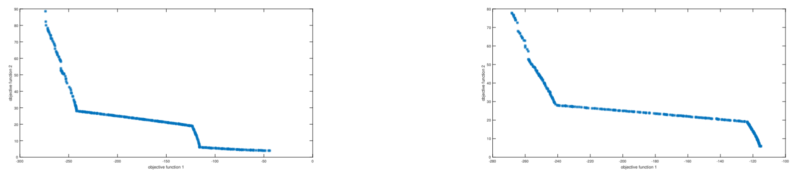


Figure 3: Flow chart of the solution procedure of FMOSPP

6. BENCHMARK INSTANCES AND COMPLEXITY OF PROPOSED METHODS

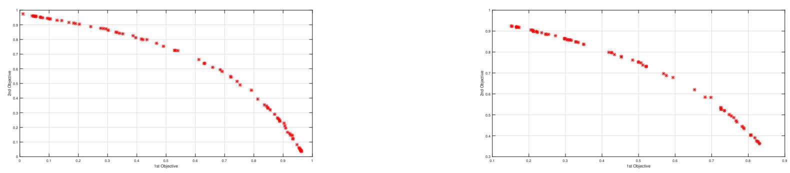
6.1. Benchmark instances

To assess the validity and effectiveness of the suggested methods, we solved some benchmark instances and compared their solutions with existing ones. The multi-objective constrained optimization test problem OSY from Deb [52] is considered to demonstrate the efficiency of AL-based NSGA-II. The AL-based NSGA-II result is compared to the previous NSGA-II result. The method suggested provides exact solutions. The only difference is that solutions below DM's AL are discarded. We have taken the DM's AL is [0.95 0.95]. The Pareto fronts of both approaches are provided in Figure 4 for comparison purposes. The FON benchmark example from Deb et al. [53] is solved to evaluate the performance of AL-based NSGA-III. For this example we have taken AL is [0.85 0.85]. The Pareto fronts for both approaches are shown in Figure 5. The proposed method provides optimal solutions that meet DM's AL.



(a) NSGA-II (b) AL-based NSGA-II

Figure 4: Pareto fronts for OSY problem [52]



(a) NSGA-III (b) AL-based NSGA-III

Figure 5: Pareto fronts for FON problem [53]

6.2. Computational Complexity

The primary criticism against multi-objective evolutionary algorithms that utilize sharing focuses and non-dominated sorting on their computational complexity, i.e., $O(MN^3)$ (where M stands for the number of objectives and N for the size of the population). NSGA-II has a computational complexity of $O(MN^2)$, as described by Deb et al. [48], whereas NSGA-III offers the same computational complexity, as stated in Deb and Jain [49]. An AL-based constraint is incorporated into NSGA-II and NSGA-III, "if" loop with a computational complexity of 1 is utilized. It does not affect the complexity of the original. Thus, the complexity of AL-based NSGA-II and AL-based NSGA-III is $O(MN^2)$.

7. NUMERICAL ILLUSTRATION FOR FMOSPP

In this section, FMOSPP is considered. This network diagram is taken from Rekh and Dhodiya [45] for solving SPP with factors time, cost, distance, and risk. This network has 10 vertices and 13 arcs, as shown in Figure 6. Every arc x_{uv} denotes the travelled time, cost, distance, and risk from vertex u to vertex v. Here time, cost, distance, and risk are in the form of TFNs (a_{uv}, b_{uv}, c_{uv}), where a_{uv}, b_{uv}, and c_{uv} represent optimistic value, most likely value, and pessimistic value respectively. The aggregated fuzzy values for criteria time, cost, distance, and risk are given in Table 1.

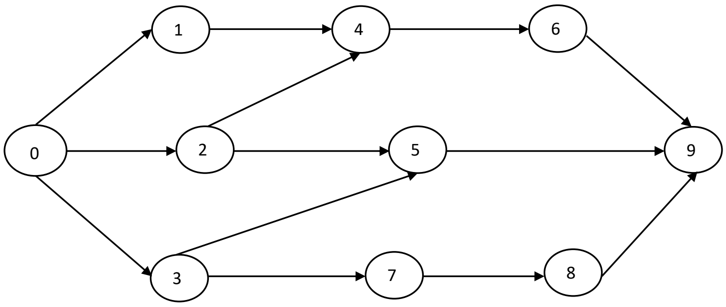


Figure 6: Network Diagram from Rekh and Dhodiya [45]

Table 1: Aggregated fuzzy values of time, cost, distance, and risk

Activity	Criteria			
	C1(Time)	C2(Cost)	C3(Distance)	C4(Risk)
0-1	(1,4.333,7)	(1200,1833.33,2400)	(0,1.67,5)	(0,2.33,5)
0-2	(1,3,5)	(400,916.67,1450)	(1,3.67,7)	(1,4.33,7)
0-3	(5,9,14)	(250,733.33,1300)	(3,7,10)	(3,6.33,9)
1-4	(3,6.667,10)	(1300,1966.67,2700)	(1,4.33,7)	(1,3.67,7)
2-4	(9,12.667,18)	(1300,1966.67,2700)	(3,6.33,9)	(5,7.67,10)
2-5	(6,9.333,12)	(5000,6166.67,7000)	(1,4.33,7)	(3,5.67,9)
3-5	(12,16,20)	(1200,2000,2800)	(1,3,5)	(1,5.67,9)
3-7	(10,13.667,17)	(1400,2066.67,2700)	(1,4.33,7)	(1,4.33,7)
4-6	(9,14,19)	(700,1166.67,1750)	(3,6.33,9)	(1,5,9)
5-9	(5,8.33,11)	(900,1466.67,2100)	(3,5.67,9)	(3,6.33,9)
6-9	(6,10.333,16)	(1400,2016.16,2600)	(5,7.67,10)	(5,8.33,10)
7-8	(13,17.667,21)	(3000,4000,5000)	(5,8.33,10)	(5,8.33,10)
8-9	(14,17,20)	(2400,3000,3600)	(5,7,9)	(5,7.67,10)

The model is coded to determine the FMOSPP solution. It is solved using MATLAB, and all tests are performed on a hp laptop equipped with an Intel(R) Core i5 10^{th} generation processor operating at 2.60 GHz and 8 GB of RAM. The following are the key characteristics for addressing the problems: The number of decision variables (13), the population size (50), the random number population generation (2000), and the iteration (20).

For $\alpha = 0$, 0.1, 0.5, and 0.9, Table 2 shows the PIS and NIS for every objective function. The exponential membership function is defined using these values. The shortest routes for FMOSPP are presented in table 8 using TPD, with varying values of the shape parameters and ALs set by the DM. Different values of α (= 0, 0.1, 0.5 & 0.9) are used here to indicate distinct scenarios of DM's confidence in fuzzy decisions. The results are calculated by estimating cases of different AL, and the different shape parameters are listed in Table 3. The notations φ, ϖ, & ϱ are from Table 8. The notation φ indicates the shortest path $0 \rightarrow 1 \rightarrow 4 \rightarrow 6 \rightarrow 9$. ϖ indicates the shortest path $0 \rightarrow 2 \rightarrow 5 \rightarrow 9$, and ϱ indicates the shortest path $0 \rightarrow 3 \rightarrow 5 \rightarrow 9$. Table 8 also indicates the corresponding objective values of the above-mentioned shortest path for $\alpha = 0, 0.1, 0.5,$ & 0.9.

Table 2: PIS and NIS values of fuzzy objectives

α-Level	Solutions	Objectives											
		z_{11}	z_{12}	z_{13}	z_{21}	z_{22}	z_{23}	z_{31}	z_{32}	z_{33}	z_{41}	z_{42}	z_{43}
0	PIS	14	22.66	30	2350	4200	6200	5	13.67	23	7	16.33	25
	NIS	42	57.33	67	7050	9800	12600	14	26.66	36	14	26.66	36
0.1	PIS	14.87	22.66	29.27	2535	4200	6400	5.87	13.67	22.07	7.93	16.33	24.13
	NIS	43.54	57.33	70.53	7325	9800	12880	15.23	26.66	35.07	15.27	26.66	35.07
0.5	PIS	18.33	22.66	26.33	3275.01	4200	7200.01	9.335	13.67	18.34	11.67	16.33	20.67
	NIS	49.67	57.33	64.66	8425.01	9800	14000	20.33	26.66	31.33	20.33	26.66	31.33
0.9	PIS	21.8	22.66	23.4	4015	4200	8000	18.9	13.67	14.6	15.4	16.33	17.2
	NIS	55.8	57.33	58.8	9525	9800	15118	3422.4	26.66	27.59	25.39	26.66	27.59

Table 3: Distinct values of shape parameters and ALs

Case	Shape parameter (K_1, K_2, K_3, K_4)	AL $(\tilde{\mu}_{z_{1j}}(X), \tilde{\mu}_{z_{2j}}(X), \tilde{\mu}_{z_{3j}}(X), \tilde{\mu}_{z_{4j}(X)})$
I	(-5,-10,-15,-20)	(0.6,0.7,0.8,0.9)
II	(-10,-15,-20,-5)	(0.7,0.8,0.9,0.6)
III	(-15,-20,-5,-10)	(0.8,0.9,0.6,0.7)
IV	(-20,-5,-10,-15)	(0.9,0.6,0.7,0.8)

7.1. The convergence rate of various methods for FMOSPP

This section discusses the case $\alpha = 0$, (-5,-10,-15,-20) shape parameter and (0.6, 0.7, 0.8, 0.9) AL.

In HGA, the above FMOSPP solution converges after 14 iterations with a population size of 50. It takes approximately 5 seconds to run the program and obtain unique optimal SP, i.e., ϱ from Table 8. This method always gives a unique solution. Figures 7, 8, 9 and 10 depict efficient solutions for time, cost, distance, and risk objective values for various shape parameter and AL combinations. In addition, these graphs depict the solution of time, cost, distance, and risk objectives as (22,33.33,45), (2350,4200,6200), (7,15.67,24) and (7,18.33,27), respectively.

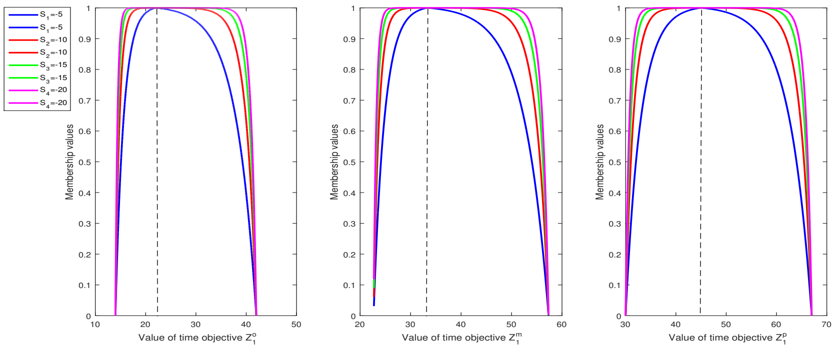


Figure 7: Efficient solution of time objective at distinct combination of shape parameter and AL

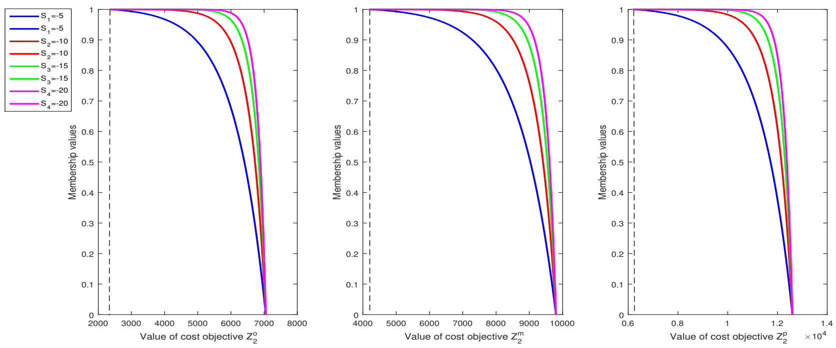


Figure 8: Efficient solution of cost objective at distinct combination of shape parameter and AL

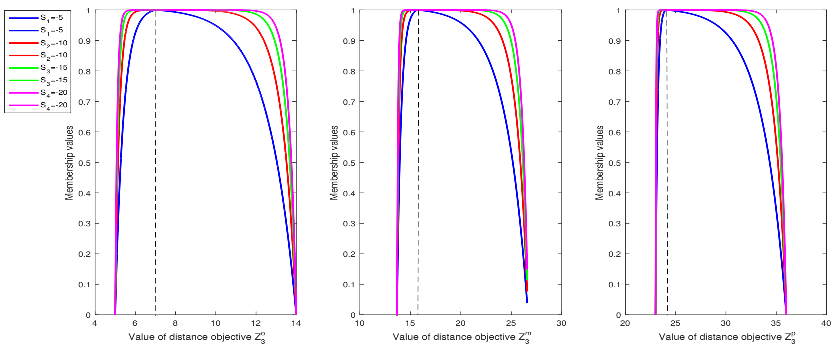


Figure 9: Efficient solution of distance objective at distinct combination of shape parameter and AL

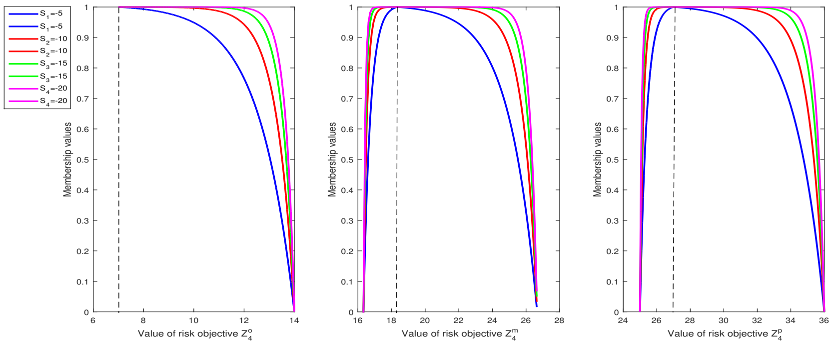


Figure 10: Efficient solution of risk objective at distinct combination of shape parameter and AL

The NSGA-II and NSGA-III approaches do not include the shape parameter and AL. The solution obtained using these approaches of the above FMOSPP for $\alpha = 0$ converges after 20 iterations with a population size of 50, and it takes approximately 10 seconds to run the program. These methods produce Pareto front with their corresponding optimal SPs, i.e., φ, ϖ, ϱ from Table 8. These methods provide the Pareto frontier.

In the proposed AL-based NSGA-II and AL-based NSGA-III approaches, the solution of the above FMOSPP is converging after 20 iterations with a population size of 50, and it takes approximately 3 seconds to run the program. A Pareto frontier is obtained, which satisfies the DM's AL with their corresponding optimal SPs, i.e., φ, ϖ, and ϱ from Table 8. These proposed methods give the Pareto frontier, which satisfies the DM's AL (i.e., the additional constraint of AL). Figures 11, 12, 13 and 14 depict the variance in goals (time, cost, distance, and risk objective) associated with distinct shape parameter preferences for $\alpha = 0$. Figures show that the resulting solutions have a more substantial effect of optimism than pessimism, reflecting the possibilistic distribution for each objective function.

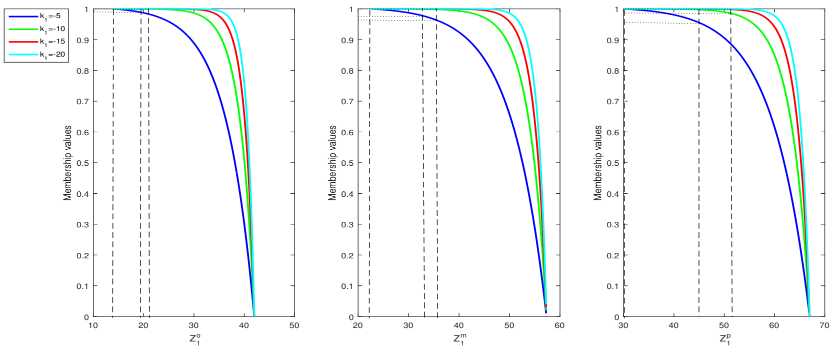


Figure 11: The level of satisfaction of the time objective for $\alpha = 0$

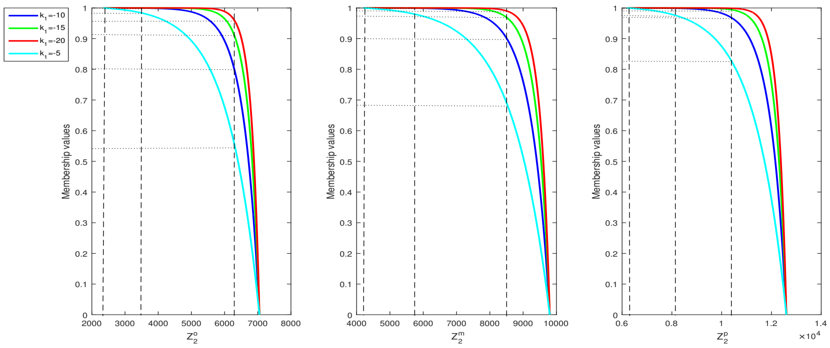


Figure 12: The level of satisfaction of the cost objective for $\alpha = 0$

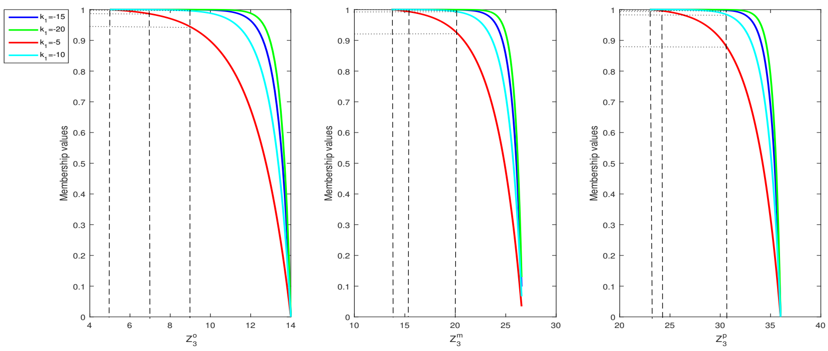


Figure 13: The level of satisfaction of the distance objective for $\alpha = 0$

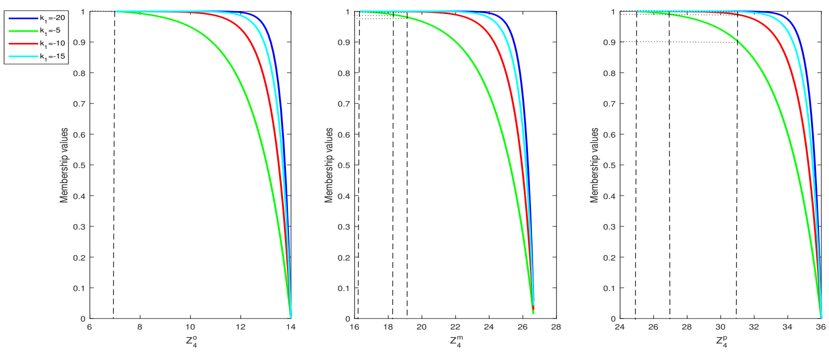


Figure 14: The level of satisfaction of the risk objective for $\alpha = 0$

7.2. Results and Discussion

For $\alpha = 0$ & case-I to case-IV of Table 3, HGA provides the solution ϱ, at the same time NSGA-II & NSGA-III give the solutions φ, ϖ, & ϱ. Proposed methods produce the solution φ, ϖ, & ϱ for cases I to III. However, in case-IV, solution ϖ does not satisfy AL of DM; therefore, in this case, these methods generate the solutions φ & ϱ. Similarly, for $\alpha = 0.1$ & case-I to case-IV of Table 3, HGA yields solution ϱ, NSGA-II & NSGA-III provide solutions φ, ϖ, & ϱ. However, proposed methods produce the solutions φ, ϖ, & ϱ for case-I to case-III while in case-IV solutions are φ & ϱ because ϖ do not satisfy AL of DM. For $\alpha = 0.5$ & case-I to case-IV of Table 3, HGA generates the solution ϱ and NSGA-II, NSGA-III & proposed methods give the solutions φ, ϖ,& ϱ. For $\alpha = 0.9$ & case-I to case-IV of Table 3, HGA provides the solution ϱ while NSGA-II, NSGA-III & proposed methods yield the solutions ϖ, & ϱ.

According to the preceding findings, the solutions found using the suggested approaches are Pareto-optimal solutions that fulfill DM's AL. As a result, the suggested approaches are best for DM concerning AL.

8. SENSITIVITY ANALYSIS

The Tables 4, 5, 6, and 7 provide the result for sensitivity analysis for given FMOSPP concerning shape parameter and AL for $\alpha = 0, 0.1, 0.5, \& 0.9$ respectively.

In Table 4, case-I, II, VII, and IX give us ϖ as a solution of given FMOSPP from proposed AL-based NSGA-II and AL-based NSGA-III. For case-III, IV, VIII, and X, the proposed approaches provide ϱ as a solution, and for case-V, VI, and XI, they generate solution φ & ϱ. Lastly, for case-XII, the above-mentioned techniques yield no solution (NaN) because there is no feasible path for that pair of shape parameters and AL for the given FMOSPP. These are the results for $\alpha = 0$, different shape parameters, and ALs. Similarly, it has been observed that the different results were obtained in Tables 5, 6, and 7.

Table 4: The results for sensitivity analysis at $\alpha = 0$

Case	Shape parameter	AL	AL-based NSGA-II	AL-based NSGA-III
			Route	Route
I	$(1, -1, 0.2, 0.5)$	$(0.5, 0.2, 0.6, 0.3)$	ϖ	ϖ
II		$(0.4, 0.2, 0.3, 0.6)$	ϖ	ϖ
III		$(0.2, 0.4, 0.6, 0.3)$	ϱ	ϱ
IV		$(0.2, 0.4, 0.3, 0.6)$	ϱ	ϱ
V	$(-8, -1, -3, -5)$	$(0.9, 0.8, 0.7, 0.6)$	φ,ϱ	φ,ϱ
VI		$(0.0, 0.3, 0.6, 0.9)$	φ, ϱ	φ, ϱ
VII		$(1.0, 0.2, 0.6, 1.0)$	ϖ	ϖ
VIII		$(0.9, 0.8, 0.75, 0.9)$	ϱ	ϱ
IX	$(0.5, -1, 5, -0.2)$	$(0.4, 0.2, 0.6, 0.3)$	ϖ	ϖ
X		$(0.2, 0.4, 0.3, 0.4)$	ϱ	ϱ
XI		$(0.2, 0.6, 0.0, 0.4)$	φ, ϱ	φ, ϱ
XII		$(0.8, 0.25, 0.3, 0.4)$	NaN	NaN

Thus, from all the previous results, it has been seen that the solution of FMOSPP changes for different shape parameters and ALs. It signifies that the solution of FMOSPP depends on the choice of shape parameter and AL. Thus, the solution of FMOSPP is sensitive concerning shape parameters and AL. Hence, DM can choose the shape parameter and AL as per the necessity of the solution.

Table 5: The results for sensitivity analysis at $\alpha = 0.1$

Case	Shape parameter	AL	AL-based NSGA-II	AL-based NSGA-III
			Route	Route
I	$(1, -1, 0.2, 0.5)$	$(0.5, 0.2, 0.6, 0.3)$	ϖ,ϱ	ϖ,ϱ
II		$(0.4, 0.2, 0.3, 0.6)$	ϖ	ϖ
III		$(0.2, 0.4, 0.6, 0.3)$	ϱ	ϱ
IV		$(0.2, 0.4, 0.3, 0.6)$	ϱ	ϱ
V	$(-8, -1, -3, -5)$	$(0.9, 0.8, 0.7, 0.6)$	φ,ϱ	φ,ϱ
VI		$(0.0, 0.3, 0.6, 0.9)$	φ, ϱ	φ, ϱ
VII		$(1.0, 0.2, 0.6, 1.0)$	ϖ	ϖ
VIII		$(0.9, 0.8, 0.75, 0.9)$	ϱ	ϱ
IX	$(0.5, -1, 5, -0.2)$	$(0.4, 0.2, 0.6, 0.3)$	ϖ	ϖ
X		$(0.2, 0.4, 0.3, 0.4)$	ϱ	ϱ
XI		$(0.2, 0.6, 0.0, 0.4)$	φ, ϱ	φ, ϱ
XII		$(0.8, 0.25, 0.3, 0.4)$	NaN	NaN

Table 6: The results for sensitivity analysis at $\alpha = 0.5$

Case	Shape parameter	AL	AL-based NSGA-II	AL-based NSGA-III
			Route	Route
I	$(1, -1, 0.2, 0.5)$	$(0.5, 0.2, 0.6, 0.3)$	ϖ,ϱ	ϖ,ϱ
II		$(0.4, 0.2, 0.3, 0.6)$	ϖ	ϖ
III		$(0.2, 0.4, 0.6, 0.3)$	ϱ	ϱ
IV		$(0.2, 0.4, 0.3, 0.6)$	ϱ	ϱ
V	$(-8, -1, -3, -5)$	$(0.9, 0.8, 0.7, 0.6)$	φ,ϱ	φ,ϱ
VI		$(0.0, 0.3, 0.6, 0.9)$	φ, ϱ	φ, ϱ
VII		$(1.0, 0.2, 0.6, 1.0)$	ϖ	ϖ
VIII		$(0.9, 0.8, 0.75, 0.9)$	φ, ϱ	φ, ϱ
IX	$(0.5, -1, 5, -0.2)$	$(0.4, 0.2, 0.6, 0.3)$	ϖ	ϖ
X		$(0.2, 0.4, 0.3, 0.4)$	ϱ	ϱ
XI		$(0.2, 0.6, 0.0, 0.4)$	φ, ϱ	φ, ϱ
XII		$(0.8, 0.25, 0.3, 0.4)$	ϖ	ϖ

Table 7: The results for sensitivity analysis at $\alpha = 0.9$

Case	Shape parameter	AL	AL-based NSGA-II	AL-based NSGA-III
			Route	Route
I	$(1, -1, 0.2, 0.5)$	$(0.5, 0.2, 0.6, 0.3)$	ϖ	ϖ,ϱ
II		$(0.4, 0.2, 0.3, 0.6)$	ϖ	ϖ
III		$(0.2, 0.4, 0.6, 0.3)$	ϱ	ϱ
IV		$(0.2, 0.4, 0.3, 0.6)$	ϱ	ϱ
V	$(-8, -1, -3, -5)$	$(0.9, 0.8, 0.7, 0.6)$	ϱ	φ,ϱ
VI		$(0.0, 0.3, 0.6, 0.9)$	φ,ϖ,ϱ	φ,ϖ,ϱ
VII		$(1.0, 0.2, 0.6, 1.0)$	ϖ	ϖ
VIII		$(0.9, 0.8, 0.75, 0.9)$	ϱ	φ, ϱ
IX	$(0.5, -1, 5, -0.2)$	$(0.4, 0.2, 0.6, 0.3)$	ϖ	ϖ
X		$(0.2, 0.4, 0.3, 0.4)$	ϱ	ϱ
XI		$(0.2, 0.6, 0.0, 0.4)$	ϱ	φ, ϱ
XII		$(0.8, 0.25, 0.3, 0.4)$	ϖ	ϖ

9. COMPARISON

Table 9 shows that HGA generates a unique SP for each case described in Table 3 at $\alpha = 0,\ 0.1,\ 0.5,\ \&\ 0.9$. Thus, it cannot find other non-dominated solutions. Consequently, NSGA-II and NSGA-III produce the Pareto front, i.e., the non-dominated solutions in the first front. Therefore it can get the solutions that are below DM's AL. According to columns 4 & 5 of Table 9, NSGA-II and NSGA-III have the same solutions for $\alpha = 0,\ 0.1,\ 0.5,\ \&\ 0.9$. Furthermore, the AL-based NSGA-II and AL-based NSGA-III generate the non-dominated solutions in the first front, which satisfy DM's AL. The SPs produced by the solution of the suggested methods are addressed in column 6 and column 7 for cases I to IV of Table 3 with $\alpha = 0,\ 0.1,\ 0.5,\ \&\ 0.9$. The solutions show that the AL-based NSGA-II and AL-based NSGA-III offers the same solutions for the specific given example. To obtain the solution of FMOSPP, HGA takes 5 seconds, NSGA-II & NSGA-III require 20 seconds, and AL-based NSGA-II & AL-based NSGA-III take 3 seconds, which is indicated in the last row of Table 9. AL-based techniques are faster than their predecessors. The methodologies suggested are the best for discovering optimal solutions in a short period concerning AL.

Therefore, the comparison, illustrated in Table 9, shows that the proposed techniques with possibility distribution can effectively tackle uncertainty in the objective function of the FMOSPP. Also, these techniques give a set of FMOSPP's non-dominated solutions based on DM's AL, choice of the shape parameter, and confidence level, which makes it easier for DM to make a decision.

10. PERFORMANCE MEASURE

The coverage performance measure [54] can be used to assess the performance of any multi-objective optimization algorithm, which is as follows::

10.1. Coverage

This performance measure compares two sets of non-dominated solutions, P and Q, and provides the percentage of individuals in one set who individuals in the other dominate. Rao [54] defines coverage as follows:

$$Cov(P,Q) = \frac{\mid \{q \in Q /\ \exists\, p \in P : p \prec = q\} \mid}{\mid Q \mid}.$$

Where P and Q are the two non-dominated sets of solutions being compared, $p \prec = q$ indicates that p dominates q or is equal to q. For $\alpha = 0$ with the case I of Table 3, coverage is calculated for the above-solved example. Consider P is a solution set of HGA method and Q is a solution set of NSGA-II, NSGA-III, AL-based NSGA-II, and AL-based NSGA-III methods one by one, then $Cov(P,Q) = \frac{1}{3},\ \forall\ Q$ and $Cov(Q,Y) = 1, \forall\ Q$. This shows that the performance of NSGA-II, NSGA-III, AL-based NSGA-II and AL-based NSGA-III is superior to that of HGA. Coverage between NSGA-II and NSGA-III solutions is $Cov(X,Y) = 1$ and $Cov(Y,X) = 1$. These values indicate that the NSGA-II and NSGA-III provide the same solutions and perform equally for this particular example. Consequently, the coverage computed for AL-based NSGA-II and AL-based NSGA-III solutions is $Cov(X,Y) = 1$ and $Cov(Y,X) = 1$. It is interpreted that AL-based NSGA-II and AL-based NSGA-III offer the same solutions and perform equally well for this specific example.

Table 8: Nomenclature of different values of α with route and its corrosponding objective values

Name	Route	Objective Value at $\alpha = 0$	Objective Value at $\alpha = 0.1$	Objective Value at $\alpha = 0.5$	Objective Value at $\alpha = 0.9$
φ	$0 \to 1 \to 4 \to 6 \to 9$	(19,35.33,52),(3500,5749.49,8050), (9,20,31) & (7,19.33,31)	(20.63,35.33,50.33),(3724.94,5749.49,8280.05.27) (10.01,20,29.09) & (8.23,19.33,29.83)	(27.17,35.33,43.67),(4624.76,5749.49,9200.27) (14.52,20,25.52) & (13.18,19.33,25.18)	(33.7,35.33,37),(5524.54,5749.49,10120.49) (18.9,20,21.1) & (18.1,19.33,20.5)
ϖ	$0 \to 2 \to 5 \to 9$	(14,22.66,30),(6300,8550.01,10550) (5,13.67,23) & (7,16.33,25)	(14.86,22.66,29.26),(6525.01,8550.01,10749.99) (5.87,13.67,20.07) & (7.93,16.33,24.13)	(18.34,22.66,26.34),(7425.02,8550.01,11550.01) (9.35,13.67,18.35) & (11.68,16.33,20.68)	(21.8,22.66,23.4),(8325,8550.01,12350) (12.8,13.67,14.6) & (15.4,16.33,17.2)
ϱ	$0 \to 3 \to 5 \to 9$	(22,33.33,45),(2350,4200,6200), (7,15.67,24) & (7,18.33,27)	(23.13,33.33,43.83),(2535,4200,6400) (7.87,15.67,23.17) & (8.13,18.33,26.13)	(27.67,33.33,39.17),(3275.01,4200,7200.01) (11.34,15.67,19.84) & (12.68,18.33,22.68)	(32.2,33.33,34.5),(4015,4200,8000) (14.8,15.67,16.5) & (17.2,18.33,19.2)

Table 9: Comparision of proposed method with existing methods

α	Case	HGA	NSGA-II	NSGA-III	AL-based NSGA-II	AL-based NSGA-III
		Route	Route	Route	Route	Route
0	I	ϱ	φ,ϖ,ϱ	φ,ϖ,ϱ	φ,ϖ,ϱ	φ,ϖ,ϱ
	II	ϱ	φ,ϖ,ϱ	φ,ϖ,ϱ	φ,ϖ,ϱ	φ,ϖ,ϱ
	III	ϱ	φ,ϖ,ϱ	φ,ϖ,ϱ	φ,ϖ,ϱ	φ,ϖ,ϱ
	IV	ϱ	φ,ϖ,ϱ	φ,ϖ,ϱ	φ,ϱ	φ,ϱ
0.1	I	ϱ	φ,ϖ,ϱ	φ,ϖ,ϱ	φ,ϖ,ϱ	φ,ϖ,ϱ
	II	ϱ	φ,ϖ,ϱ	φ,ϖ,ϱ	φ,ϖ,ϱ	φ,ϖ,ϱ
	III	ϱ	φ,ϖ,ϱ	φ,ϖ,ϱ	φ,ϖ,ϱ	φ,ϖ,ϱ
	IV	ϱ	φ,ϖ,ϱ	φ,ϖ,ϱ	φ,ϱ	φ,ϱ
0.5	I	ϱ	φ,ϖ,ϱ	φ,ϖ,ϱ	φ,ϖ,ϱ	φ,ϖ,ϱ
	II	ϱ	φ,ϖ,ϱ	φ,ϖ,ϱ	φ,ϖ,ϱ	φ,ϖ,ϱ
	III	ϱ	φ,ϖ,ϱ	φ,ϖ,ϱ	φ,ϖ,ϱ	φ,ϖ,ϱ
	IV	ϱ	φ,ϖ,ϱ	φ,ϖ,ϱ	φ,ϖ,ϱ	φ,ϖ,ϱ
0.9	I	ϱ	ϖ,ϱ	ϖ,ϱ	ϖ,ϱ	ϖ,ϱ
	II	ϱ	ϖ,ϱ	ϖ,ϱ	ϖ,ϱ	ϖ,ϱ
	III	ϱ	ϖ,ϱ	ϖ,ϱ	ϖ,ϱ	ϖ,ϱ
	IV	ϱ	ϖ,ϱ	ϖ,ϱ	ϖ,ϱ	ϖ,ϱ
run time:		5 seconds	20 seconds	20 seconds	3 seconds	3 seconds

11. CONCLUSION

In order to develop solutions for the FMOSPP, the suggested AL-based NSGA-II and AL-based NSGA-III techniques have optimized the optimistic, most likely, and pessimistic scenarios of fuzzy objective functions with TPD using the EMF with specific, realistic constraints. The suggested approaches have solved a numerical example with ten vertices and thirteen arcs. AL-based NSGA-II and AL-based NSGA-III offer the same solutions within a short time other than previous methods for the provided example. For this above-solved specific example, the performance of both methods is similar. However, AL-based NSGA-III is faster than AL-based NSGA-II concerning time to get an optimal solution. Additionally, the DM now has greater flexibility for ALs, shape parameters, and effective SPs. This study concluded that AL-based NSGA-II and AL-based NSGA-III are appropriate for DM concerns with AL.

The proposed study investigated MOSPP in a fuzzy environment. This problem can also be extended to other uncertain environments, such as uncertain environments, uncertain interval environments, type-2 uncertain environments, etc. The suggested approaches can be used to address a variety of optimization problems, such as those relating to transportation, travelling salesmen, assignments, etc.

Acknowledgements: Authors would like to express their sincere gratitude to the anonymous reviewers who gave us insight that made this article more significant.

Funding: This research received no external funding.

REFERENCES

[1] R. Bellman, "On a routing problem," *Quarterly of applied mathematics*, vol. 16, no. 1, pp. 87–90, 1958.

[2] E. W. Dijkstra *et al.*, "A note on two problems in connexion with graphs," *Numerische mathematik*, vol. 1, no. 1, pp. 269–271, 1959.

[3] R. W. Floyd, "Algorithm 97: shortest path," *Communications of the ACM*, vol. 5, no. 6, p. 345, 1962.

[4] S. E. Dreyfus, "An appraisal of some shortest-path algorithms," *Operations research*, vol. 17, no. 3, pp. 395–412, 1969.

[5] P. Hansen, "Bicriterion path problems," in *Multiple criteria decision making theory and application*. Springer, 1980, pp. 109–127.

[6] E. Q. V. Martins, "On a multicriteria shortest path problem," *European Journal of Operational Research*, vol. 16, no. 2, pp. 236–245, 1984.

[7] P. Serafini, "Some considerations about computational complexity for multi objective combinatorial problems," in *Recent advances and historical development of vector optimization*. Springer, 1987, pp. 222–232.

[8] X. Gandibleux, F. Beugnies, and S. Randriamasy, "Martins' algorithm revisited for multi-objective shortest path problems with a maxmin cost function," *4OR*, vol. 4, no. 1, pp. 47–59, 2006.

[9] Z. Tarapata, "Selected multicriteria shortest path problems: An analysis of complexity, models and adaptation of standard algorithms," *International Journal of Applied Mathematics and Computer Science*, vol. 17, no. 2, p. 269, 2007.

[10] J. C. Clímaco and M. M. Pascoal, "Multicriteria path and tree problems: discussion on exact algorithms and applications," *International Transactions in Operational Research*, vol. 19, no. 1-2, pp. 63–98, 2012.

[11] A. Sedeno-Noda and A. Raith, "A dijkstra-like method computing all extreme supported non-dominated solutions of the biobjective shortest path problem," *Computers & Operations Research*, vol. 57, pp. 83–94, 2015.

[12] A. Sedeno-Noda and M. Colebrook, "A biobjective dijkstra algorithm," *European Journal of Operational Research*, vol. 276, no. 1, pp. 106–118, 2019.

[13] L. D. P. Pugliese, J. Granat, and F. Guerriero, "Two-phase algorithm for solving the preference-based multicriteria optimal path problem with reference points," *Computers & Operations Research*, vol. 121, p. 104977, 2020.

[14] C. Ntakolia and D. K. Iakovidis, "A swarm intelligence graph-based pathfinding algorithm (sigpa) for multi-objective route planning," *Computers & Operations Research*, vol. 133, p. 105358, 2021.

[15] P. M. de las Casas, A. Sedeño-Noda, and R. Borndörfer, "An improved multiobjective shortest path algorithm," *Computers & Operations Research*, p. 105424, 2021.

[16] D. J. Dubois, *Fuzzy sets and systems: theory and applications*. Academic press, 1980, vol. 144.

[17] S. Okada and T. Soper, "A shortest path problem on a network with fuzzy arc lengths," *Fuzzy sets and systems*, vol. 109, no. 1, pp. 129–140, 2000.

[18] I. Mahdavi, R. Nourifar, A. Heidarzade, and N. M. Amiri, "A dynamic programming approach for finding shortest chains in a fuzzy network," *Applied Soft Computing*, vol. 9, no. 2, pp. 503–511, 2009.

[19] A. Tajdin, I. Mahdavi, N. Mahdavi-Amiri, and B. Sadeghpour-Gildeh, "Computing a fuzzy shortest path in a network with mixed fuzzy arc lengths using α-cuts," *Computers & Mathematics with Applications*, vol. 60, no. 4, pp. 989–1002, 2010.

[20] S. Mukherjee, "Fuzzy programming technique for solving the shortest path problem on networks under triangular and trapezoidal fuzzy environment," *International Journal of Mathematics in Operational Research*, vol. 7, no. 5, pp. 576–594, 2015.

[21] A. Ebrahimnejad, Z. Karimnejad, and H. Alrezaamiri, "Particle swarm optimisation algorithm for solving shortest path problems with mixed fuzzy arc weights," *International Journal of Applied Decision Sciences*, vol. 8, no. 2, pp. 203–222, 2015.

[22] A. Ebrahimnejad, M. Tavana, and H. Alrezaamiri, "A novel artificial bee colony algorithm for shortest path problems with fuzzy arc weights," *Measurement*, vol. 93, pp. 48–56, 2016.

[23] G. Sheng, Y. Su, and W. Wang, "A new fractal approach for describing induced-fracture porosity/permeability/compressibility in stimulated unconventional reservoirs," *Journal of Petroleum Science and Engineering*, vol. 179, pp. 855–866, 2019.

[24] Y. Yang, J. Hu, Y. Liu, and X. Chen, "Alternative selection of end-of-life vehicle management in china: A group decision-making approach based on picture hesitant fuzzy measurements," *Journal of cleaner production*, vol. 206, pp. 631–645, 2019.

[25] H. Zhao, L. Xu, Z. Guo, W. Liu, Q. Zhang, X. Ning, G. Li, and L. Shi, "A new and fast waterflooding optimization workflow based on insim-derived injection efficiency with a field application," *Journal of Petroleum Science and Engineering*, vol. 179, pp. 1186–1200, 2019.

[26] M. Enayattabr, A. Ebrahimnejad, H. Motameni, and H. Garg, "A novel approach for solving all-pairs shortest path problem in an interval-valued fuzzy network," *Journal of Intelligent & Fuzzy Systems*, vol. 37, no. 5, pp. 6865–6877, 2019.

[27] M. Enayattabar, A. Ebrahimnejad, and H. Motameni, "Dijkstra algorithm for shortest path problem under interval-valued pythagorean fuzzy environment," *Complex & Intelligent Systems*, vol. 5, no. 2, pp. 93–100, 2019.

[28] A. Ebrahimnejad, S. Tabatabaei, and F. J. Santos-Arteaga, "A novel lexicographic optimization method for solving shortest path problems with interval-valued triangular fuzzy arc weights," *Journal of Intelligent & Fuzzy Systems*, vol. 39, no. 1, pp. 1277–1287, 2020.

[29] A. Ebrahimnejad, M. Enayattabr, H. Motameni, and H. Garg, "Modified artificial bee colony algorithm for solving mixed interval-valued fuzzy shortest path problem," *Complex & Intelligent Systems*, vol. 7, no. 3, pp. 1527–1545, 2021.

[30] A. Abbaszadeh Sori, A. Ebrahimnejad, and H. Motameni, "Elite artificial bees' colony algorithm to solve robot's fuzzy constrained routing problem," *Computational Intelligence*, vol. 36, no. 2, pp. 659–681, 2020.

[31] A. Ebrahimnejad, "An acceptability index based approach for solving shortest path problem on a network with interval weights," *RAIRO-Operations Research*, vol. 55, pp. S1767–S1787, 2021.

[32] L. Lin, C. Wu, and L. Ma, "A genetic algorithm for the fuzzy shortest path problem in a fuzzy network," *Complex & Intelligent Systems*, vol. 7, no. 1, pp. 225–234, 2021.

[33] D. Di Caprio, A. Ebrahimnejad, H. Alrezaamiri, and F. J. Santos-Arteaga, "A novel ant colony algorithm for solving shortest path problems with fuzzy arc weights," *Alexandria Engineering Journal*, vol. 61, no. 5, pp. 3403–3415, 2022.

[34] G. V. Rani and B. Reddy, "Multi-objective fuzzy shortest path selection for green routing and scheduling problems." *International Journal of Advanced Research in Computer Science*, vol. 8, no. 7, 2017.

[35] Y. Zhang, P. Liu, L. Yang, and Y. Gao, "A bi-objective model for uncertain multi-modal shortest path problems," *Journal of Uncertainty Analysis and Applications*, vol. 3, no. 1, pp. 1–17, 2015.

[36] M. Bagheri, A. Ebrahimnejad, S. Razavyan, F. Hosseinzadeh Lotfi, and N. Malekmohammadi, "Fuzzy arithmetic dea approach for fuzzy multi-objective transportation problem," *Operational Research*, pp. 1–31, 2020.

[37] S. Ghosh, S. K. Roy, A. Ebrahimnejad, and J. L. Verdegay, "Multi-objective fully intuitionistic fuzzy fixed-charge solid transportation problem," *Complex & Intelligent Systems*, vol. 7, no. 2, pp. 1009–1023, 2021.

[38] M. Bagheri, A. Ebrahimnejad, S. Razavyan, F. H. Lotfi, and N. Malekmohammadi, "Solving fuzzy multi-objective shortest path problem based on data envelopment analysis approach," *Complex & Intelligent Systems*, vol. 7, no. 2, pp. 725–740, 2021.

[39] A. Abbaszadeh Sori, A. Ebrahimnejad, and H. Motameni, "The fuzzy inference approach to solve multi-objective constrained shortest path problem," *Journal of Intelligent & Fuzzy Systems*, vol. 38, no. 4, pp. 4711–4720, 2020.

[40] R.-C. Wang and T.-F. Liang, "Applying possibilistic linear programming to aggregate production planning," *International journal of production economics*, vol. 98, no. 3, pp. 328–341, 2005.

[41] P. Gupta and M. K. Mehlawat, "A new possibilistic programming approach for solving fuzzy multiobjective assignment problem," *IEEE Transactions on Fuzzy Systems*, vol. 22, no. 1, pp. 16–34, 2013.

[42] L. A. Zadeh, "Fuzzy sets," in *Fuzzy sets, fuzzy logic, and fuzzy systems: selected papers by Lotfi A Zadeh*. World Scientific, 1996, pp. 394–432.

[43] Y.-J. Lai and C.-L. Hwang, "A new approach to some possibilistic linear programming problems," *Fuzzy sets and systems*, vol. 49, no. 2, pp. 121–133, 1992.

[44] A. R. Tailor and J. M. Dhodiya, "Genetic algorithm based hybrid approach to solve optimistic, most-likely and pessimistic scenarios of fuzzy multi-objective assignment problem using exponential membership function," *Journal of Advances in Mathematics and Computer Science*, pp. 1–19, 2016.

[45] R. K. Rekh and J. M. Dhodiya, “Solution of fuzzy multi criteria project management problem by fuzzy programming technique with possibilistic approach,” *Indian J. Sci. Technol*, vol. 12, pp. 1–21, 2019.

[46] N. Srinivas and K. Deb, “Muiltiobjective optimization using nondominated sorting in genetic algorithms,” *Evolutionary computation*, vol. 2, no. 3, pp. 221–248, 1994.

[47] D. Gordberg, *Genetic Algorithm in Search, Optimization and Machine Learning*. Addison-Wesley, Readingm MA, 1989.

[48] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan, “A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: Nsga-ii,” in *International conference on parallel problem solving from nature*. Springer, 2000, pp. 849–858.

[49] K. Deb and H. Jain, “An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: solving problems with box constraints,” *IEEE transactions on evolutionary computation*, vol. 18, no. 4, pp. 577–601, 2013.

[50] K. Deb, S. Bandaru, and H. Seada, “Generating uniformly distributed points on a unit simplex for evolutionary many-objective optimization,” in *International Conference on Evolutionary Multi-Criterion Optimization*. Springer, 2019, pp. 179–190.

[51] I. Das and J. E. Dennis, “Normal-boundary intersection: A new method for generating the pareto surface in nonlinear multicriteria optimization problems,” *SIAM journal on optimization*, vol. 8, no. 3, pp. 631–657, 1998.

[52] K. Deb, *Multi-Objective Optimization using Evolutionary Algorithms*. Wiley India Pvt. Limited, 2010. ISBN 9788126528042. [Online]. Available: https://books.google.co.in/books?id=Cph6cgAACAAJ

[53] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, “Scalable test problems for evolutionary multiobjective optimization,” in *Evolutionary multiobjective optimization*. Springer, 2005, pp. 105–145.

[54] R. V. Rao, *Jaya: an advanced optimization algorithm and its engineering applications*. Springer, 2019.