Original Scientific Article Journal of Decision Analytics and Intelligent Computing
Vol. 3 issue 1, (2023) 122-138 https://doi.org/10.31181/jdaic10015082023t

Bi-objective covering salesman problem with uncertainty

Siba Prasada Tripathy[1,*]

[1] Department of CSE, Silicon Institute of Technology, Bhubaneswar, India

* Correspondence: siba.dilu@gmail.com

Received 11 March 2023

Accepted for publication 8 August 2023

Published 15 August 2023

Abstract

Humanitarian relief transportation and mass fatality management activities are the most strenuous tasks after a natural or artificial disaster. A feasible and realistic transport model is essential for accomplishing the tasks in a planned way. Covering Salesman Problem (CSP) is a variant of Traveling Salesman Problem (TSP) which has been used in many application areas, including disaster management. In this paper, we consider a bi-objective CSP in an uncertain environment where Interval Type 2 fuzzy numbers represent the costs of the edges. A new local search technique is introduced in the memetic algorithm, which has been used to solve the problem. A computational experiment on a set of instances indicates the effectiveness of the introduced local search technique along with the proposed methodology.

Keywords: covering salesman problem, memetic algorithm, local search, uncertainty, fuzzy numbers

1. Introduction

Humanitarian relief transportation and mass fatality management activities can be planned successfully with a feasible and realistic transport model. These two tasks can be modeled using Covering Salesman Problem (CSP), and hence, in the recent past, researchers have paid huge attention to design methodologies for solving different variants of CSP (Current and Schilling, 1989; Jensen, 1999; Shaelaie et al., 2014). Here, CSP can be defined as given a set of vertices in a given graph (V, E), the goal of CSP is to minimize the cost of the tour by starting the journey from a node (i.e., depot), traversing a subset of vertices (i.e., facilities) which cover the given number of vertices (i.e., customers) which is shown in Figure 1. A customer is covered, if it comes within a pre-defined distance of a facility point on the tour. Uncertainty, again, is unavoidable while solving real world problems. Therefore, designing methodologies for CSP in an uncertain environment is also an important task. In this work, using interval type-2 fuzzy set (IT2FS), we propose a methodology for transforming the uncertain CSP to a deterministic one and then solving using the memetic algorithm.

After a rigorous study on Traveling Salesman Problem (TSP) (Shaelaie et al., 2014), many people have contributed to covering salesman problem (CSP). The covering salesman problem (CSP) (Current and Schilling,

1989) is a variant of traveling salesman problem where the goal is to find a tour of minimum length cycle traversing a subset of vertices (i.e., nodes in a graph) such that every other node which is not included in the tour and is bounded by a pre-determined distance (radius) from at least one visited node. Many methodologies have been developed to solve CSP, but all the models need more realistic approaches for performing relief activities after a disaster, rural health care delivery and emergency vaccination. Aftermath a disaster, it is seen that the relief activities have been conducted without proper planning and design of transportation model, resulting in a huge demise of lives. In humanitarian relief transportation and mass fatality management activities (Jensen, 1999), an efficient transportation model for tackling such strenuous activities can save many lives. Due to a lack of time and adequate resources, it is difficult for the relief team to reach all the places. Hence, traversing a subset of the places with gathering nearby people/victims who lie within a pre-determined distance makes the process easy to accomplish. Here, a place is considered a node or city. The CSP can also be applied in post-disaster mass fatality management, humanitarian relief transportation, rural health care delivery, telecommunication network (Jensen, 1999), etc. The travelling plan usually considers more than one objective considering distance and survivability, both attributes.

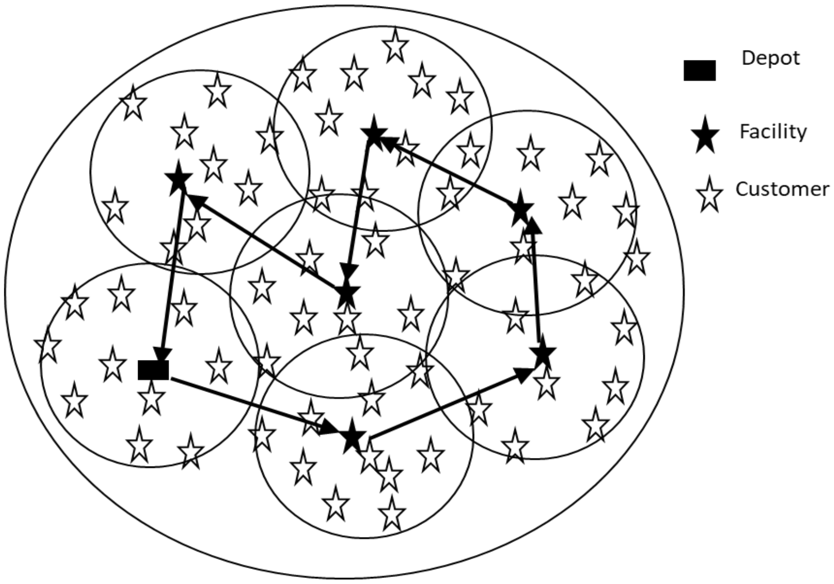

Figure 1. An example path of Covering Salesman Problem

In this paper, we have suggested an approach more realistic than existing models of CSP. Here, we have considered the distance between two nodes and the survivability point of the location after travelling from a particular place. The distance and survivability scenario may come after a natural disaster where the distance and survivability are sometimes conflicting.

The rest of the paper is organized as follows. The preliminaries have been presented in Section 2. The literature study on various variants of CSP has been presented in Section 3. Section 4 contains the problem formulation and mathematical model of the proposed problem. The methodology is given in Section 5. Section 6 contains the results and analysis of the results. Finally, the work has been concluded in Section 7.

2. Literature Review

The covering salesman problem was initially introduced in 1980 through the thesis of Current, and in 1989, the first heuristic for the Constraint Satisfaction Problem (CSP) was proposed. Current and Schilling (1989) presented a heuristic approach to tackle the covering salesman problem). This problem aims to find the shortest visiting cycle that covers a subset of nodes within a predefined distance. The research was extended in Arkin and Hassin (1994) where the geometric constraints of the covering salesman problem were demonstrated. The study includes bounded error ratio analysis and polynomial time approximation algorithms. To overcome the limitations of integer programming highlighted in Patterson and Rolland (2003), the authors suggested creating several sub-graphs from a main graph.

Numerous studies have explored various variants of the Traveling Salesman Problem (TSP), such as the pickup-and-delivery TSP (Zhao et al., 2009), multi-depot multiple TSP (Ghafurian and Javadian, 2011), online TSP (Wen et al., 2012), clustered TSP (Bao and Liu, 2012)., generalized TSP (Laporte, 1996; Karapetyan and Gutin, 2012), and more. These TSP variants are extensively discussed and described in Gutin and Punnen (2006). In addition, the Covering Tour Problem (CTP) (Gendreau et al., 1997) is also addressed in the research, considering different class constraints. A branch and cut algorithm with polyhedral properties is employed to tackle CTP, even for large instances with up to 600 vertices. The algorithm classifies the vertices into three groups: S1, S2, and S3. Group S1 comprises the vertices visited by the tour, group S2 contains the vertices covered by the tour, and group S3 represents the remaining vertices that are neither visited nor covered by the tour. Furthermore, the algorithm incorporates the Lin–Kernighan heuristic (Lin and Kernighan, 1973) to improve the arrangement of the visited vertices.

A heuristic algorithm utilizing Integer Linear Programming (ILP) was created for the CSP (Wu and Mendel, 2011). The algorithm starts with an initial feasible solution and employs a destroy-and-repair approach to enhance the tour length. By removing certain vertices from the tour and reassigning them, a new feasible solution is formed through the optimization of an ILP-based model. The results demonstrate that the ILP-based method exhibits greater efficiency than other algorithms for solving the CSP.

A CSP variation was suggested in Golden et al. (2012), where the generalisations of CSP are reviewed while relaxing the visiting or covering of vertices. More than one vertices may be visited. The authors considered several rural healthcare delivery models that include overnight stays. Shaelaie et al. (2014) put out two mathematical solutions to the problem of covering a specified number of nodes. The Memetic Algorithm (MA) and Variable Neighbourhood Search (VNS) meta-heuristic algorithms have also been suggested, and their findings have been compared with those of simulations of mathematical models and results from current data using IBM CPLEX.

Salari et al. (2015) developed a hybrid heuristic approach for CSP that blends Ant Colony Optimisation (ACO) with a CSP dynamic programming heuristic. The starting node in this technique is a dummy vertex that does not cover any other nodes, and the tour concludes by covering a subset of nodes to supply the required demand. Ozbaygin et al. (2016) designed a branch and cut technique to maximize coverage demand within a time limit, and the demand of the vertex is fully covered if it is part of the tour and half covered if it is not part of the tour but close to the traversed vertex.

A green vehicle routing problem has been solved using a revised intelligent algorithm where the goal is to use less fuel while completing the tour (Wang et al., 2019). The work Du et al. (2019) has drawn a conclusion about making healthy travel for tourism which is a recent thrust point for many researchers. Similar works have been done by some researchers where different domains of travelling salesman problems and vehicle routing problems have been discussed (Ni et al., 2017; Avila-Torres et al., 2018; Rainer et al., 2018; Wu et al., 2019) and also in Zhang et al. (2019), fuzzy logic has been used for multi-attribute decision making.

Another procedure for addressing different objectives can be handled with the weighted sum method, as described in Kim and Weck (2004) and Stanimirovic et al. (2011). Through the utilization of this method, every point obtained represents a Pareto optimal solution for the multi-objective optimization problem at hand. The weights employed within this approach hold essential importance as they influence preferences, the Pareto optimal set, and objective-function values. Previous research Marler and Arora (2010) has identified the factors that determine the resulting solution point when specific weights are used. Furthermore, significant drawbacks have been recognized concerning the upfront expression of preferences, leading to guidelines that prevent the method's indiscriminate application.

The work Tripathy et al. (2021) proposes a novel variant of the classical travelling salesman problem (TSP), which is called the multi-objective covering salesman problem with 2-coverage (MOCSP-2). In MOCSP-2, the objective is to find a minimum cost Hamiltonian cycle that visits all the nodes in the graph while satisfying two additional constraints. First, each node should be covered by at least two different cycles, and second, the length of the longest cycle should be minimized. The authors propose a mathematical model for MOCSP-2 and show that it is an NP-hard problem. In another work, Biswas et al. (2022), the authors propose a mathematical model for MOCS, which involves multiple objectives, including minimizing the cost, minimizing the number of cycles, and maximizing the coverage of nodes.

For IT2FSs, ranking techniques, similarity and uncertainty metrics are crucial ideas. Authors from different fields have offered numerous ranking systems. Karnik and Mendel (2001) have suggested a centroid-based ranking approach. For solving fuzzy multiple attributes group decision-making situations with greater adaptability and intelligence, Lee and Chen (2008) introduced a new ranking mechanism. This ranking system, which is based on trapezoidal IT2FSs, has been utilised for this work.

A recent study by Qin and Liu (2015) introduced fundamental concepts and operational laws regarding Interval Type-2 Fuzzy Sets (IT2FSs). The study also presented three types of ranking value formulas for evaluating IT2FSs using arithmetic average (AA), geometric average (GA), and harmonic average (HA) operators. Additionally, the desirable properties of these ranking value formulas were discussed. Building upon these properties, the notion of combined ranking value was introduced, and a new interval type-2 fuzzy entropy was further developed. The new entropy measure employed the trigonometric sine function to quantify the uncertainty present in IT2FSs.

3. Mathematical Modeling

Mathematical model and the notations for CSP with radius are given below:

$G = (V, A)$: A directed graph

V: Set of vertices

A: Set of edges

W: Set of customers

F: Set of Facilities

(i, j): An edge between the vertices *i* and *j*

c_{ij} : Cost (or length) of the edge(i, j)

d_j: Demand covered by the facility *j*

D: Total demand

λ_{ij}: Continuous variable presenting the load of the tour

$$\delta_{ij} = \begin{cases} 1 \text{ if edge } (i,j) \text{ is visited by the tour,} \\ 0 \; otherwise \end{cases} \quad \forall (i,j) \in A \tag{1}$$

$$z_{ij} = \begin{cases} 1 \text{ if the customer } i \text{ is assigned to the facility j,} \\ 0 \; otherwise \end{cases} \quad \forall (i,j) \in A \tag{2}$$

$$\Delta_i = \begin{cases} 1 \text{ if vertex } i \text{ is visited,} \\ 0 \; otherwise \end{cases} \quad \forall (i) \in V \tag{3}$$

$$Minimize \sum_{(i,j)\in A} c_{ij}\delta_{ij} \,, Minimize \sum_{(i,j)\in A} t_{ij}\delta_{ij} \tag{4}$$

Subject to:

$$\sum_{\forall\, j \in F\cup\{0\}} z_{ij} = 1 \quad \forall i \in W \tag{5}$$

$$\sum_{\forall i\in W} d_i z_{ij} = D \quad j \in F \cup \{0\} \tag{6}$$

$$\sum_{j\in F\cup\{0\}} \delta_{0j} = 1 \tag{7}$$

$$\sum_{j\in F\cup\{0\}} \delta_{ij} = \sum_{j\in F\cup\{0\}} \delta_{ji} \quad \forall\, i \in F \cup \{0\} \tag{8}$$

$$z_{ij} \le \sum_{j\in F\cup\{0\}} \delta_{kj} + \sum_{j\in F\cup\{0\}} \delta_{jk} \quad \forall\, i \in W,\ j \in F,\ i = 1 \dots n \tag{9}$$

$$\sum_{j\in F} \lambda_{0j} = \sum_{i\in W} \sum_{j\in F\cup\{0\}} z_{ij} \tag{10}$$

$$\sum_{j\in F\cup\{0\}} \lambda_{ji} - \sum_{j\in F\cup\{0\}} \lambda_{ij} = \sum_{j\in F\cup\{0\}} \delta_{ji} \quad \forall\, i \in F \cup \{0\} \tag{11}$$

$$\sum_{j\in F} \lambda_{j0} = 0 \tag{12}$$

$$\delta_{ij} \in \{0,1\} \quad \forall\, i, \in F \cup \{0\} \tag{13}$$

$$z_{ij} \in \{0,1\} \quad \forall\, i \in W,\ \forall\, j \in F_i \tag{14}$$

$$\lambda_{ij} \ge 0 \quad \forall (i,j) \in F \cup \{0\} \tag{15}$$

The objective function (4) is to minimize the tour distance and survivability. Once a customer is assigned to a facility, then that customer cannot be assigned to any other facility, even if it lies within the radius of other facilities. Constraint (5) ensures that each customer $i \in W$ is assigned only to one facility $j \in F$. Total demand D must be met by the facilities as shown in constraint (6). Constraint (7) ensures that the tour must start from the depot and the in-degree and out-degree constraints for each $i \in F \cup \{0\}$ are represented by the constraint (8). Constraint (9) shows that the customer i is allocated to facility j, if j is visited by the tour. λ_{ij} represents the pre-specified tour load which is the total number of nodes to be covered by the tour. Each node is assumed to be a single point of load. Hence, tour load is the pre-specified value D at the depot and it gradually decreases, finally making it zero when the last facility point is met with the depot. Constraints (10), (11) and (12) model the sub-tour elimination constraints. Model variables are depicted in (13) to (15).

4. Methodology

We have developed a memetic algorithm and applied it to the randomly created travel distance and survivability during travel between cities (algorithm 1). The memetic algorithm has used a basic evolutionary algorithm structure combined with local search techniques. It includes the GA steps with 2-opt (Lin and Kernighan, 1973) and dropS-and-add (drop a segment) as two local search techniques. We propose the dropS-and-add approach in this work. We have considered the cost in terms of crisp values and then applied our memetic algorithm with interval type-2 fuzzy numbers in uncertain environments. While computing the fitness calculation, the bi-objective version of CSP, considering a weighted sum model, we considered having two objectives: cost and survivability which are presented below.

$$Minimize\ \ F = w_1 f_1 + \ w_1 f_1 \tag{16}$$

Where

- F is the total value out of two objectives,
- w_1 and w_2 are the weights for two objectives and $w_1 + \ w_2 = 1$,
- f_1 = Cost as first objective,
- f_2 = Survivability as second objective.

Algorithm 1. Memetic Algorithm

1. *Set the parameter values*
2. *Construct initial population*
3. *Evaluate each fitness value.*
4. *while termination is false do*
5. *Selection*
6. *Crossover operation.*
8. *Mutation operation.*
9. *Apply local search on 10% chromosome of the population*
10. *Evaluate each fitness value.*
11. *end*

4.1. Parameter Setting

Parameter values are given in Table 1, where N_p represents the population size, T_N represents the number of nearest uncovered customers to be considered while choosing the facility for the next visit, L_c is the percentage of chromosomes where the local search method has been applied. If there is no improvement continuously for λ times, the algorithm shall be terminated. For the proposed algorithm, P_c is the crossover probability and P_m is the mutation rate. w_1 and w_2 are weights taken for accomplishing both the cost and the survivability factor in weighted sum model.

Table 1. Parameter values used in proposed multi-objective algorithm

N_p	T_N	P_c	P_m	L_c	Λ
200	5	0.8	0.07	10	50

4.2. Initial population

Here, the first facility is taken randomly. To choose the next facility, $T_N$number of uncovered nodes are considered, out of which one node is randomly selected as the next facility point to be visited. These T_N number of nodes are considered from the uncovered nodes. The method of constructing initial pool is given below:

4.3. Fitness evaluation

The total fitness of the path or chromosome is equal to the salesman's total distance travelled. The length of each edge visited by the tour is added to determine each chromosome's fitness.

4.4. Selection

Two types of selection techniques have been followed: binary tournament and Roulette wheel selection method for selecting the pool of chromosomes for crossover. 50% of the population are taken using binary tournament selection method and remaining 50% is taken using roulette wheel selection method.

4.5. Crossover

A new crossover operator called GPX operator (Tripathy et al., 2017) has been used in this algorithm. Here, we assume a Global Parent (*GP*) chromosome. *GP* contains all the nodes as genes available for a particular data set of the problem. Table 2 shows the GP for the TSP instance eil51 containing 51 nodes.

Table 2. *GP* of the instance eil51

0	1	2	…	50

An example of crossover is presented in Figure 2.

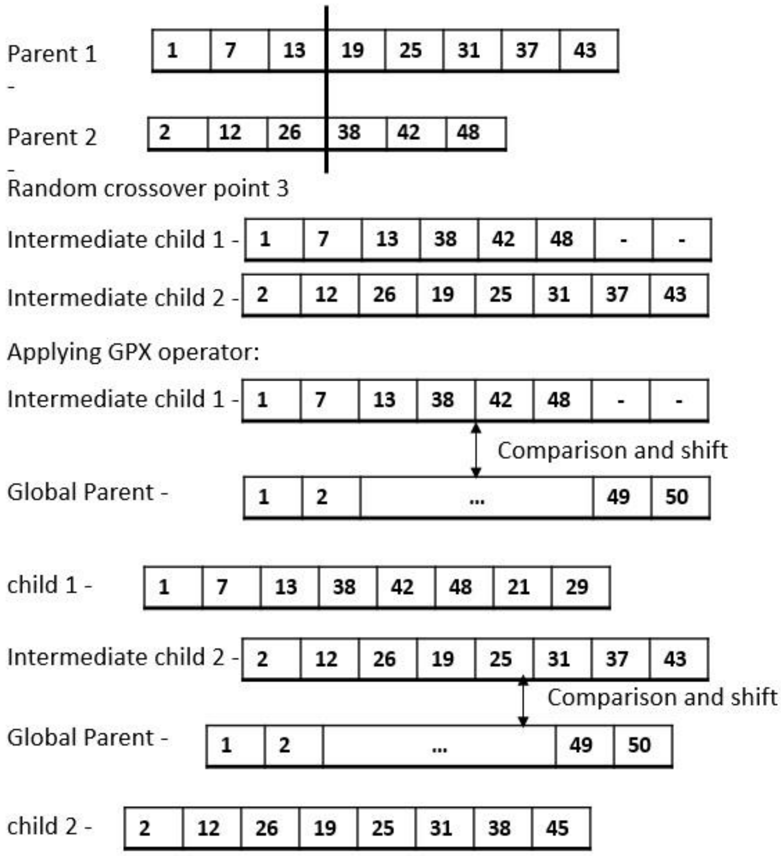

Figure 2. Example of crossover

4.6. Mutation

For mutation, two random facility points are taken and they swap their positions. After swapping if it gives the better result, this change is made and continues the process with mutation rate P_m. This is called induced mutation and it can reduce the tour length after mutation.

4.7. LS-Heuristic

We proposed a local search heuristic with two different types of local search techniques for the memetic algorithm. For both techniques, we have considered L_c percentage of chromosomes of the population size.

This heuristic includes the 2-opt method and dropS-and-add technique. We have proposed the DropS-and-add method where we drop two consecutive facility points and try to re-insert other facility points to reduce the cost. If the cost is reduced then we update the path, otherwise, we discard it. 2-opt (Lin and Kernighan, 1973) procedure is considered where two edges are removed and other two edges are inserted in the tour. If the cost of the new tour is less than the previous then the process is retained, otherwise, it is rolled back. The local search techniques are used in this problem to maintain a balance between diversity and convergence. The detail of the LS-heuristic and the proposed dropS-ad-add are presented in Algorithm 2 and Algorithm 3.

Algorithm 2. LS-Heuristic

```
1.repeat
2.      start again:
3.      best cost  =calculate cost(existing path)
4.      modified path = dropS-and-add(existing path)
5.      nodes-swap = total number of nodes eligible for swapping.
6.      i = 0, j = 0
7.      while i<=  nodes-swapdo
8.              while j  <=  nodes-swapdo
9.              new path =  2-opt(modified path, i, j)
10.             end while
11.     end while
12.     new cost =  calculate cost(new path)
13.     ifnew cost<best costthen
14.             best cost  =new cost
15.     endif
16.     goto start again
17.  until no improvement is made
```

Algorithm 3. DropS and add

```
1.      Remove a vertex point x from the given path (existing path)
2.      Covered-point = Covered-pointU a, where a∈ covered points by x.
3.      Remove another vertex point y from the existing path, consecutive to the previous point.
4.      Covered-point = Covered-pointU b, where b∈ covered points by y
5.      repeat
6.      Take point p and add it to the existing path in the place where the first point was dropped, where
p∈Covered-point
7.              while all the nodes have not been covered yet
8.                      Take point q, where q∈Covered-point
9.                      Add the point q in the existing path next after the point p
10.             end while
11.     until all the points ∈Covered-point has been taken
```

4.8. Uncertainty handling with IT2FS

All the previous works on CSP are based on minimizing the tour length. In this paper, we have considered two objectives of the problem. First is the distances between the cities (i.e., nodes) as interval type 2 fuzzy numbers for the uncertainty issue and the survivability of the victims have been taken to include another corresponding objective of the problem.

For comparison between two IT2F numbers we have used their rank values. Centroid ranking method (Mendel, 2011) has been used to find the value of rank.

An IT2FS is defined by the reference points and the corresponding heights of the upper and lower membership functions. In our algorithm, we took trapezoidal IT2FS into account. Figure 3 above depicts a trapezoidal IT2FS A. The FOU is the area that is shaded. Both a lower membership function (LMF)$\underline{\mu}_i(x_i)$ and an upper membership function (UMF) $\overline{\mu}_i(x_i)$ define its boundaries. They are both type-1 fuzzy sets (T1FSs), the UMF and LMF.

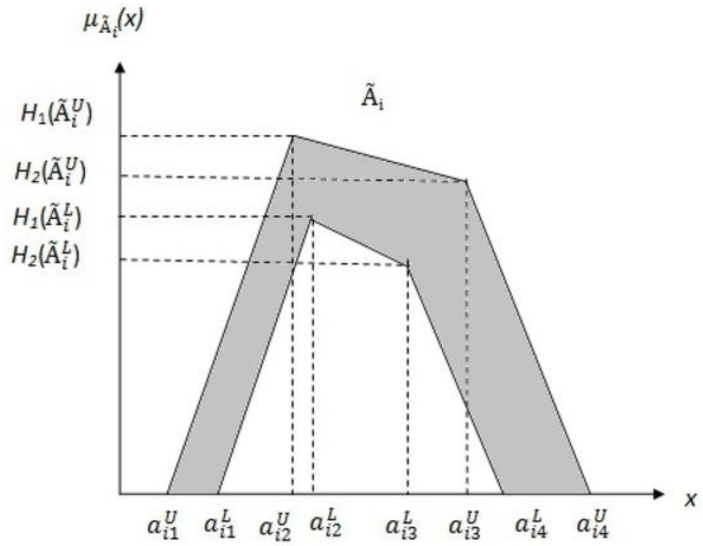

Figure 3. Twelve reference points for determining FOU and ($a^U{}_{i1}$,$a^U{}_{i2}$, $a^U{}_{i3}$, $a^U{}_{i4}$) represent a trapezoidal UMF $a^U{}_i$, with height $H_1(\tilde{a}_i^U)$ and $H_2(\tilde{a}_i^U)$and ($a^L{}_{i1}$, $a^L{}_{i1}$, $a^L{}_{i1}$, $a^L{}_{i1}$) determines a trapezoidal LMF $a^U{}_i$ with height $H_1(\tilde{a}_i^L)$and $H_1(\tilde{a}_i^L)$

Addition: Addition between two fuzzy numbers has been done in the following manner.

$$\tilde{A}_1 = ((a^U{}_{11}, a^U{}_{12}, a^U{}_{13}, \dots \dots . . a^U{}_{1n}); \left(H_1(\tilde{a}_1^U), H_2(\tilde{a}_1^U), \dots \dots H_{n-1}(\tilde{a}_1^U)\right))$$
$$((a^L{}_{11}, a^L{}_{12}, a^L{}_{13}, \dots \dots . . a^L{}_{1n}); \left(H_1(\tilde{a}_1^L), H_2(\tilde{a}_1^L), \dots \dots H_{n-1}(\tilde{a}_1^L)\right)) \quad (17)$$

$$\tilde{A}_2 = ((a^U{}_{21}, a^U{}_{22}, a^U{}_{23}, \dots \dots . . a^U{}_{2n}); \left(H_1(\tilde{a}_2^U), H_2(\tilde{a}_2^U), \dots \dots H_{n-1}(\tilde{a}_2^U)\right))$$
$$((a^L{}_{21}, a^L{}_{22}, a^L{}_{23}, \dots \dots . . a^L{}_{2n}); \left(H_1(\tilde{a}_1^L), H_2(\tilde{a}_1^L), \dots \dots H_{n-1}(\tilde{a}_1^L)\right)) \quad (18)$$

$$\tilde{A}_1 \oplus \tilde{A}_2 = ((a^U{}_{11} + a^U{}_{21}, a^U{}_{12} + a^U{}_{12} \dots \dots \dots a^U{}_{1n} + a^U{}_{2n}); \left(min\left(H_1(\tilde{a}_1^U), H_1(\tilde{a}_2^U)\right), \dots . \min\left(H_{n-1}(\tilde{a}_1^U), H_{n-1}(\tilde{a}_1^U)\right)\right),$$
$$(a^L{}_{11} + a^L{}_{21}, a^L{}_{12} + a^L{}_{12} \dots \dots \dots a^L{}_{1n} + a^L{}_{2n});$$
$$(min\left(H_1(\tilde{a}_1^L), H_1(\tilde{a}_2^L)\right), \dots . \min\left(H_{n-1}(\tilde{a}_1^L), H_{n-1}(\tilde{a}_2^L)\right)) \quad (19)$$

Let's see an example of the addition operation with two trapezoidal fuzzy numbers:

$\tilde{A}_1 = (0.49, 0.77, 0.77, 0.87;\ 1, 1), (0.67, 0.77, 0.77, 0.85;\ 0.95, 0.95)$

$\tilde{A}_2 = (0.27, 0.52, 0.52, 0.64;\ 1, 1), (0.4, 0.52, 0.52, 0.64;\ 0.95, 0.95)$

$$\tilde{A}_1 \oplus \tilde{A}_2 = ((0.49 + 0.27), (0.77 + 0.52), (0.77 + 0.52), (0.87 + 0.64);\ \min(1,1), \min(1,1)), ((0.67 + 0.4), (0.77 + 0.52), (0.77 + 0.52), (0.85 + 0.64);\ \min(0.95, 0.95), \min(0.95, 0.95))$$
$$= (0.76, 1.26, 1.26, 1.51;\ 1, 1), (1.07, 1.29, 1.29, 1.49;\ 0.95, 0.95)$$

$$\tilde{A}_1 \oplus \tilde{A}_2 \oplus \tilde{A}_3 \oplus \tilde{A}_4 \oplus \dots \dots \dots \tilde{A}_n = ((\tilde{A}_1^U, \tilde{A}_1^L) \oplus (\tilde{A}_2^U, \tilde{A}_2^L) \oplus \dots \dots \dots \dots \oplus (\tilde{A}_n^U, \tilde{A}_n^L))$$
$$= ((\tilde{A}_1^U \oplus \tilde{A}_2^U \oplus \tilde{A}_3^U \dots \dots \dots .. \tilde{A}_n^U), (\tilde{A}_1^L \oplus \tilde{A}_2^L \oplus \tilde{A}_3^L \dots \dots \dots .. \tilde{A}_n^L)$$
$$= ((A_{11}^U + A_{21}^U + A_{31}^U + \dots + A_{n1}^U, \quad A_{12}^U + A_{22}^U + A_{32}^U + \dots + A_{n2}^U, \quad \dots \dots \dots \dots, \quad A_{1n}^U + A_{2n}^U + A_{3n}^U + \dots + A_{nn}^U;\ \min\left(H_1(\tilde{a}_1^U), H_1(\tilde{a}_2^U), \dots \dots H_1(\tilde{a}_n^U)\right), \min\left(H_2(\tilde{a}_1^U), H_2(\tilde{a}_2^U), \dots \dots H_2(\tilde{a}_n^U)\right)) \ (A_{11}^L + A_{21}^L + A_{31}^L + \dots + A_{n1}^L, \quad A_{12}^L + A_{22}^L + A_{32}^L + \dots + A_{n2}^L, \quad \dots \dots \dots \dots, \quad A_{1n}^L + A_{2n}^L + A_{3n}^L + \dots + A_{nn}^L;\ \min\left(H_1(\tilde{a}_1^L), H_1(\tilde{a}_2^L), \dots \dots H_1(\tilde{a}_n^L)\right), \min\left(H_2(\tilde{a}_1^L), H_2(\tilde{a}_2^L), \dots \dots H_2(\tilde{a}_n^L)\right)))$$

Ranking: Ranking of IT2FSs is an important role in the decision-making process with IT2FS.

How to compare two IT2FSs is important since, unlike real numbers, they do not naturally form a linear order. Intervals have been utilized with type-2 fuzzy lengths. Basically, it is a way to convert an IT2FS into a real number. The actual figures are compared next. The literature has a variety of ranking techniques (Lee and Chen, 2008; Mendel, 2011; Chiao, 2012). The centroid ranking method (Karnik and Mendel, 2001) is employed in this work to compare the IT2FSs. According to Karnik and Mendel (2001), c_l and c_r are calculated from the higher membership functions of A in the manner described below:

$$c_l = min_{L \epsilon N} centroid\left(A_e(L)\right), \tag{20}$$

$$c_r = min_{R \epsilon N} centroid\left(A_e(R)\right), \tag{21}$$

Where

$$centroid\left(A_e(L)\right) = \frac{\sum_{i=1}^{L} x_i \overline{\mu}_{\tilde{A}}(x_i) + \sum_{i=L+1}^{N} x_i \underline{\mu}_{\tilde{A}}(x_i)}{\sum_{i=1}^{L} \overline{\mu}_{\tilde{A}}(x_i) + \sum_{i=L+1}^{N} \underline{\mu}_{\tilde{A}}(x_i)} \tag{22}$$

$$centroid\left(A_e(R)\right) = \frac{\sum_{i=1}^{R} x_i \underline{\mu}_{\tilde{A}}(x_i) + \sum_{i=R+1}^{N} x_i \overline{\mu}_{\tilde{A}}(x_i)}{\sum_{i=1}^{R} \underline{\mu}_{\tilde{A}}(x_i) + \sum_{i=R+1}^{N} \overline{\mu}_{\tilde{A}}(x_i)} \tag{23}$$

where $L \in N$ is the switch point that makes the change from $\overline{\mu}_{\tilde{A}}$ to $\underline{\mu}_{\tilde{A}}$ (Figure 4(a)), and $R \in N$ is the switch point which marks the changes from $\underline{\mu}_{\tilde{A}}$ to $\overline{\mu}_{\tilde{A}}$ (Figure 4 (b)), N is the number of discrete points on which the $x -$ domain of A has been credited. The pseudo-code for computing L and c_l is given in algorithm below. We considered at the discrete version of the algorithm here.

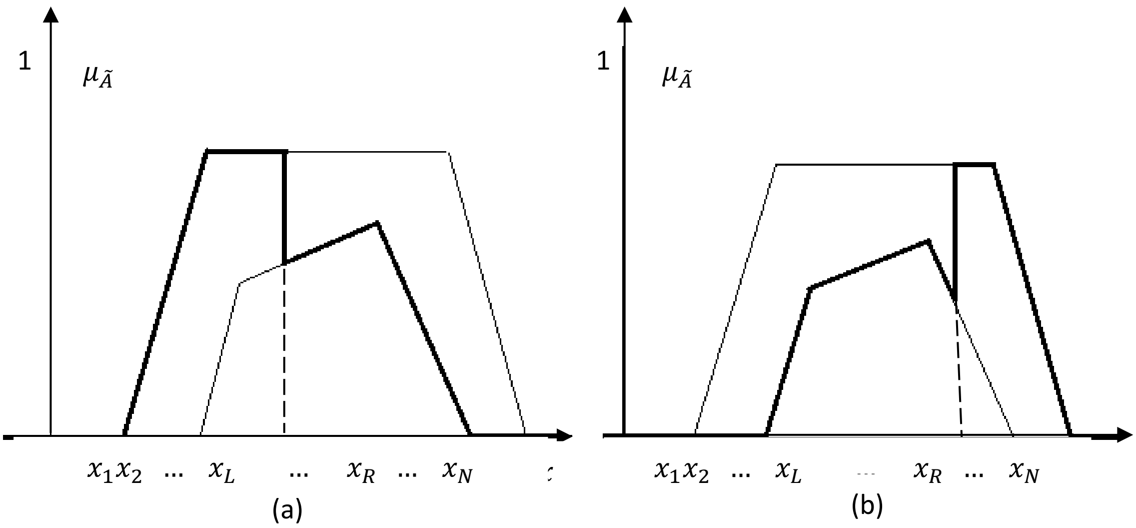

Figure 4. (a) Interpolation of switch point L (b) Interpolation of switch point R

The discrete version of KM algorithm and recursive algorithm:
In order to find L, and consequently c_l, the KM algorithm goes as follows:

1. Start the search by computing an initial point $\acute{c}$:

$$\acute{c} = \frac{\sum_{i=1}^{N} x_i \theta_i}{\sum_{i=1}^{N} \theta_i} \tag{24}$$

With $\theta_i = (\underline{\mu}_{\tilde{A}}(x_i) + \overline{\mu}_{\tilde{A}}(x_i))/2\,,\quad i = 1, 2, 3, \dots \dots \dots, N$ (25)

2. Find k $(1 \le k \le N - 1)$ such that $x_k \le \acute{c} \le x_{k+1}$.

3. Set $\theta_i = \begin{cases} \underline{\mu}_{\tilde{A}}(x_i), & i \le k, \\ \overline{\mu}_{\tilde{A}}(x_i), & i > k, \end{cases}$ and compute $\acute{\acute{c}} = \frac{\sum_{i=1}^{N} x_i \theta_i}{\sum_{i=1}^{N} \theta_i}$ (26)

4. If $\acute{c} = \acute{\acute{c}}$ then stop and set $c_l = \acute{\acute{c}}$, $L = k$. Else go to step 5.

5. Set$\acute{c} = \acute{\acute{c}}$ and go to step 2.

Then, the average of c_l and c_r of IT2FS $\tilde{A}$, i.e., $c(\tilde{A})$ is computed using the following formula, which is the centroid-based ranking value of IT2FS$\tilde{A}$.

$$c(\tilde{A}) = \frac{(c_l + c_r)}{2} \tag{27}$$

The larger is the centroid value $c(\tilde{A})$, the greater is the arc length corresponding IT2FS $\tilde{A}$, that is less acceptable, as we have minimization objective. As an example, let us consider the IT2FS $\tilde{A} = ((5.38, 7.50, 9.00, 9.81; 1, 1), (8.29, 8.56, 8.56, 9.21; 0.38, 0.38))$. To find the centroid-based rank of $\widetilde{A_{1,2}}$, we consider N, specified in approach, as 50. We computed the initial point $\grave{c}_l = 7.86$ using (4.8.10). This value is assigned to $\grave{c}_l$. Then the value of $k = 21$ is computed following the loop defined in step 2. From this, we get $\grave{\grave{c}}_l = 7.28$ using the value k as 21. As the values of $\grave{c}_l (= 7.86)$ and $\grave{\grave{c}}_l (= 7.28)$ are not equal, we repeat our process (step 2 to step 4) of the algorithm until the value of $\grave{c}_l$ and $\grave{\grave{c}}_l$ converges, i.e., they have the same value, which is 7.24 for this example. So, we get the value of $c_l = 7.24$. Similarly, we compute the value of $c_r = 8.56$. The average of c_l and c_r of IT2FS $\widetilde{A_{12}}$, i.e., $c\left(\widetilde{A_{12}}\right) = 7.90$ is computed using (4.8.13)

Let $\tilde{A}$ and $\tilde{B}$ are two IT2FSs. Then

$a)\ \tilde{A}\ greater\ than\ \tilde{B}$ if and only if $c(\tilde{A}) < c(\tilde{B})$.

$b)\ \tilde{A}\ less\ than\ \tilde{B}$ if and only if $c(\tilde{A}) > c(\tilde{B})$.

$c)\ \tilde{A}n\ equivalent\ to\ \tilde{B}$ if and only if $c(\tilde{A}) = c(\tilde{B})$.

Here, $\tilde{A}\ greater\ than\ \tilde{B}$ implies that the length of the arc which is represented by $\tilde{A}$ is less than the length of the arc represented by $\tilde{B}$. Similarly, $\tilde{A}\ equivalent\ to\ \tilde{B}$ implies that the length of the arc represented by $\tilde{A}$is same the length of the arc represented by $\tilde{B}$. $\tilde{A}\ less\ than\ \tilde{B}$ implies that the length of the arc represented by $\tilde{A}$is greater than the length of the arc represented by $\tilde{B}$. As a result, the ranking values of IT2FSs establish a natural order among those IT2F numbers.

5. Results and Discussion

We have implemented the proposed memetic algorithm with both types of distance parameters (discrete and fuzzy) for the distance between cities. To find the covered customers, we have used the *NC* nearest customers with a fixed value of NC. We have used 15 instances, such as distance and survivability instances, which ranges from 50 nodes to 130 nodes. We have used the system of Intel® Core (TH) 2 Duo CPU with E8400@3.00 Ghz having 4 GB RAM for the execution of the algorithms. We have implemented the algorithms in JAVA in Net Beans IDE 8.0. To calculate the tour cost, we have taken three different values of *NC* measurements, which show the flexibility in selecting the coverage area. As per the requirement, *NC* can be changed when developing the transport model after a disaster. The instances for the problem are generated randomly. For the instances for the deterministic version of CSP, we have named it as DI and followed by the number of nodes. For uncertain CSP, we have named as UI and followed by the number of nodes. Here Di is denoted as a deterministic instance and UI is denoted as an uncertain instance. The number of nodes varies from 50 to 130 in the instances for both versions of CSP.

In Table 3, we have shown the results of the memetic algorithm with discrete values of the distance between cities. Here the weights are 0.5 for distance and 0.5 for survivability. Best-obj. represents the best objective value in five times of its execution, F_c represents the number of facilities visited in the tour, $T_v$is the survivability to visit the tour for the best cost in seconds. $Best - obj_{dist}$for the total tour distance for the Best-obj where $w1 = 1$ and $w2 = 0$and $Best - obj_{comfort}$ for storing total tour survivability of the Best-obj of the results having $w1 = 0$ and $w2 = 1$with corresponding dataset and NC values.

In Table 4 above, we have shown the results of the memetic algorithm with IT2FS values of the distance between cities and considering weights 0.5 and 0.5, respectively. The survivability instances are the same as the previous discrete method. Best-obj. represents the best objective value in hundred times of its execution, F_c represents the number of facilities visited in the tour, T_v is the survivability after visiting the tour for the best cost in seconds with corresponding dataset and NC values.

Visualization of the weighted sum module: In the result section, we have taken the weight factors for distance and survivability as 0.5 and 0.5. That means we have weighted both the data equally and formed the instances DI and UI. The solution that passes through the black line will be the non-dominated solution. The solutions in the graph are shown in pair i.e., [weight of distance] [weight of survivability].

Table 3. Results of the multi-objective CSP with discrete distance data set's value

Name	Memetic Algorithm					
	NC	Best-obj	F_c	T_v	$Best - obj_{dist}$	$Best - obj_{comfort}$
DI50	7	280	11	5.04	226	334
	8	263	9	2.23	187	339
	9	247	8	1.54	212	282
DI55	7	317	12	4.67	272	362
	8	282	10	1.83	272	293
	9	272	9	3.08	318	226
DI60	7	361	13	4.19	338	384
	8	334	12	5.11	247	421
	9	310	11	1.83	285	335
DI70	7	382	17	5.51	341	423
	8	342	14	3.88	316	367
	9	326	13	3.49	350	301
DI80	7	464	19	8.80	424	505
	8	428	19	3.57	349	506
	9	397	14	6.42	398	395
DI85	7	440	18	15.79	411	470
	8	429	19	4.02	380	479
	9	403	16	3.73	343	463
DI90	7	457	18	10.92	442	473
	8	444	17	9.15	439	449
	9	407	16	6.82	364	450
DI95	7	493	20	8.53	403	582
	8	458	19	18.38	461	454
	9	420	18	8.28	393	448
DI100	7	535	23	10.19	480	589
	8	493	20	15.42	439	547
	9	466	19	6.28	458	475
DI105	7	586	25	20.56	506	667
	8	509	20	13.22	499	519
	9	483	19	16.53	395	572
DI110	7	593	25	11.82	544	642
	8	537	24	8.98	440	635
	9	507	19	20.08	404	609
DI115	7	576	24	21.67	491	661
	8	563	24	25.57	477	649
	9	510	22	14.74	406	615
DI120	7	619	30	20.68	521	717
	8	574	23	16.98	486	661
	9	526	20	24.44	493	559
DI125	7	639	26	55.26	625	653
	8	599	25	17.51	535	663
	9	557	23	43.86	523	591
DI130	7	671	32	23.62	583	758
	8	612	27	17.14	480	744
	9	591	24	15.63	409	773

Table 4. Results of the multi-objective CSP with IT2FS distance data set's value

Name	Memetic Algorithm			
	NC	Best-obj	F_c	T_v
UI50	7	341	8	2.15
	8	323	7	2.23
	9	282	6	1.27
UI55	7	405	9	2.03
	8	340	7	1.98
	9	325	7	2.70
UI60	7	440	10	2.26
	8	400	9	2.31
	9	386	8	4.06
UI70	7	513	12	4.90
	8	451	11	2.73
	9	433	10	1.97
UI80	7	574	13	7.09
	8	553	12	9.97
	9	524	12	2.77
UI85	7	625	15	9.97
	8	547	13	6.09
	9	521	12	6.71
UI90	7	681	17	5.51
	8	596	14	4.28
	9	584	13	10.90
UI95	7	700	17	12.16
	8	645	16	5.39
	9	611	14	5.14
UI100	7	751	18	12.58
	8	688	16	11.06
	9	602	14	4.64
UI105	7	807	19	14.21
	8	767	18	12.35
	9	692	16	7.31
UI110	7	827	20	19.93
	8	778	19	7.60
	9	719	17	7.04
UI115	7	884	21	16.24
	8	774	18	12.20
	9	741	18	10.22
UI120	7	892	21	18.40
	8	827	20	19.78
	9	773	19	9.84
UI125	7	930	22	30.79
	8	858	21	11.20
	9	815	19	10.11
UI130	7	992	25	30.42
	8	923	23	26.70
	9	836	19	16.34

The above figure 5 clearly shows that with each weight distribution, the problem gives the non-dominated solution with the best solutions taken from each weight group. The best solutions offer good diversity. Now it's upon the user to select a particular weight per requirement.

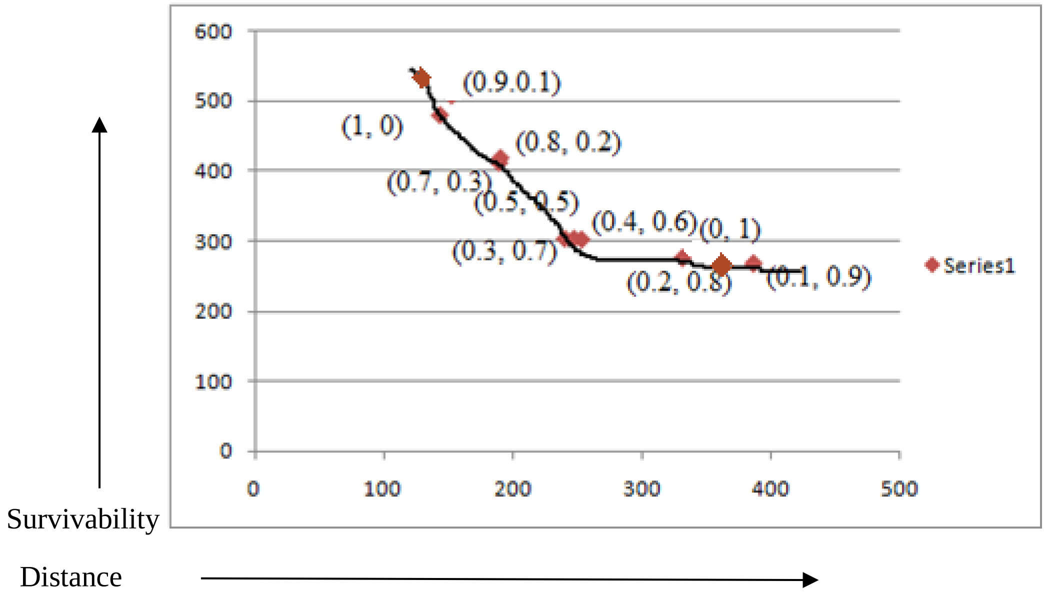

Figure 5. Plotting of the best solutions in each weight combinations.

6. Conclusions

The goal of the CSP problem is to cover all the victims in a devastated area after a disaster by traversing a set of facility points where any victim or customer can come and get the service. Any customer point can be treated as a facility in this problem. Here, a multi-objective version of the deterministic covering salesman problem is formulated and solved. We have developed two memetic algorithms introducing a new local search technique to solve the problem. In addition to this, we have extended the deterministic CSP towards uncertainty. We have used interval type-2 fuzzy numbers as the distance between cities to handle the uncertainty. This type of problem can be used specifically in disaster management, rural health care delivery, mass fatality management, etc. The results indicate that the memetic algorithm with introduced local search outperforms the other memetic algorithm. The work can be extended in the future by taking more than two objectives and having more uncertainty.

References

Arkin, E.M., & Hassin, R. (1994). Approximation algorithms for the geometric covering salesman problem. Discrete Applied Mathematics, 55(3), 197-218.

Avila-Torres, P., Caballero, R., Litvinchev, I., Lopez-Irarragorri, F., & Vasant, P. (2018). The urban transport planning with uncertainty in demand and travel time: a comparison of two defuzzification methods. Journal of Ambient Intelligence and Humanized Computing, 9(3), 843-856.

Bao, X., & Liu, Z. (2012). An improved approximation algorithm for the clustered traveling salesman problem. Information Processing Letters, 112(23), 908-910.

Biswas, A., Tripathy, S. P., & Pal, T. (2022). On multi-objective covering salesman problem. Neural Computing and Applications, 34(24), 22127–22140.

Chiao, K.P. (2012). Trapezoidal interval type-2 fuzzy set extension of analytic hierarchy process. 2012 IEEE International Conference on Fuzzy Systems (pp. 1-8). Brisbane, QLD, Australia: IEEE.

Current, J. R., & Schilling, D. A. (1989). The covering salesman problem. Transportation science, 23(3), 208-213.

Du, S., Zhang, H., Xu, H., Yang, J., & Tu, O. (2019). To make the travel healthier: a new tourism personalized route recommendation algorithm. Journal of Ambient Intelligence and Humanized Computing, 10, 3551–3562.

Gendreau, M., Laporte, G., & Semet, F. (1997). The covering tour problem. Operations Research, 45(4), 568-576.

Ghafurian, S., & Javadian, N. (2011). An ant colony algorithm for solving fixed destination multi-depot multiple traveling salesmen problems. Applied Soft Computing, 11(1), 1256-1262.

Golden, B., Naji-Azimi, Z., Raghavan, S., Salari, M., & Toth, P. (2012). The generalized covering salesman problem. INFORMS Journal on Computing, 24(4), 534-553.

Gutin, G., & Punnen, A. P. (Eds.). (2006). The traveling salesman problem and its variations (Vol. 12). Springer Science & Business Media.

Jensen, R. A. (1999). Mass fatality and casualty incidents: a field guide. CRC Press.

Karapetyan, D., & Gutin, G. (2012). Efficient local search algorithms for known and new neighborhoods for the generalized traveling salesman problem. European Journal of Operational Research, 219(2), 234-251.

Karnik, N.N., & Mendel, J.M. (2001). Centroid of a type-2 fuzzy set. Information Sciences, 132(1-4), 195-220.

Kim, I.Y., & de Weck, O. (2004). Adaptive weighted sum method for multiobjective optimization. In 10th AIAA/ISSMO multidisciplinary analysis and optimization conference (p. 4322). Albany, New York: American Institute of Aeronautics and Astronautics.

Laporte, G., Asef-Vaziri, A., & Sriskandarajah, C. (1996). Some applications of the generalized travelling salesman problem. Journal of the Operational Research Society, 47(12), 1461-1467.

Lee, L.W., & Chen, S.M. (2008). A new method for fuzzy multiple attributes group decision-making based on the arithmetic operations of interval type-2 fuzzy sets. International Conference on Machine Learning and Cybernetics, vol 6, (pp. 3084-3089). Kunming, China: IEEE.

Lin, S., & Kernighan, B.W. (1973). An effective heuristic algorithm for the traveling-salesman problem. Operations research, 21(2), 498-516.

Marler, R.T., & Arora, J. S. (2010). The weighted sum method for multi-objective optimization: new insights. Structural and multidisciplinary optimization, 41(6), 853-862.

Mendel, J. M. (2011). On centroid calculations for type-2 fuzzy sets. Applied and Computational Mathematics, 10(1), 88-96.

Ni, Y., Shi, Q., & Wei, Z. (2017). Optimizing influence diffusion in a social network with fuzzy costs for targeting nodes. Journal of Ambient Intelligence and Humanized Computing, 8(5), 819-826.

Ozbaygin, G., Yaman, H., & Karasan, O.E. (2016). Time constrained maximal covering salesman problem with weighted demands and partial coverage. Computers & Operations Research, 76, 226-237.

Patterson, R., & Rolland, E. (2003). The cardinality constrained covering traveling salesman problem. Computers & Operations Research, 30(1), 97-116.

Qin, J., & Liu, X. (2015). Multi-attribute group decision making using combined ranking value under interval type-2 fuzzy environment. Information Sciences, 297, 293-315.

Rainer, J. J., Cobos-Guzman, S., & Galán, R. (2018). Decision making algorithm for an autonomous guide-robot using fuzzy logic. Journal of Ambient Intelligence and Humanized Computing, 9(4), 1177-1189.

Salari, M., Reihaneh, M., & Sabbagh, M.S. (2015). Combining ant colony optimization algorithm and dynamic programming technique for solving the covering salesman problem. Computers & Industrial Engineering, 83, 244-251.

Shaelaie, M.H., Salari, M., & Naji-Azimi, Z. (2014). The generalized covering traveling salesman problem. Applied Soft Computing, 24, 867-878.

Stanimirovic, I.P., Zlatanovic, M.L., & Petkovic, M.D. (2011). On the linear weighted sum method for multi-objective optimization. Facta Universitatis, Series: Mathematics and Informatics, 26(4), 49-63.

Tripathy, S. P., Biswas, A., & Pal, T. (2021). A multi-objective covering salesman problem with 2-coverage. Applied Soft Computing, 113, 108024.

Tripathy, S.P., Tulshyan, A., Kar, S., & Pal, T. (2017). A Metameric Genetic Algorithm with New Operator for Covering Salesman Problem with Full Coverage. International Journal of Control Theory and Applications, 10(7), 245-252.

Wang, R., Zhou, J., Yi, X., & Pantelous, A.A. (2019). Solving the green-fuzzy vehicle routing problem using a revised hybrid intelligent algorithm. Journal of Ambient Intelligence and Humanized Computing, 10(1), 321-332.

Wen, X., Xu, Y., & Zhang, H. (2012). Online traveling salesman problem with deadline and advanced information. Computers & Industrial Engineering, 63(4), 1048-1053.

Wu, D., & Mendel, J. M. (2011). On the continuity of type-1 and interval type-2 fuzzy logic systems. IEEE Transactions on Fuzzy Systems, 19(1), 179-192.

Wu, L., Ma, W., Yang, Y., & Wang, K. (2019). A competitive analysis approach for route choice with uncertain travel times and blocked nodes. Journal of Ambient Intelligence and Humanized Computing, 10(1), 345-355.

Zhang, C., Wang, C., Zhang, Z., & Tian, D. (2019). A novel technique for multiple attribute group decision making in interval-valued hesitant fuzzy environments with incomplete weight information. Journal of Ambient Intelligence and Humanized Computing, 10(6), 2417-2433.

Zhao, F., Li, S., Sun, J., & Mei, D. (2009). Genetic algorithm for the one-commodity pickup-and-delivery traveling salesman problem. Computers & Industrial Engineering, 56(4), 1642-1648.