

Research Article

Application of Fuzzy Optimization to the Orienteering Problem

Madhushi Verma and K. K. Shukla

Department of Computer Science and Engineering, IIT (BHU), Varanasi 221005, India

Correspondence should be addressed to Madhushi Verma; madhushi.rs.cse@itbhu.ac.in

Received 5 February 2015; Accepted 18 June 2015

Academic Editor: Katsuhiko Honda

Copyright © 2015 M. Verma and K. K. Shukla. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper deals with the orienteering problem (OP) which is a combination of two well-known problems (i.e., travelling salesman problem and the knapsack problem). OP is an NP-hard problem and is useful in appropriately modeling several challenging applications. As the parameters involved in these applications cannot be measured precisely, depicting them using crisp numbers is unrealistic. Further, the decision maker may be satisfied with graded satisfaction levels of solutions, which cannot be formulated using a crisp program. To deal with the above-stated two issues, we formulate the *fuzzy* orienteering problem (FOP) and provide a method to solve it. Here we state the two necessary conditions of OP of maximizing the total collected score and minimizing the time taken to traverse a path (within the specified time bound) as fuzzy goals and the remaining necessary conditions as crisp constraints. Using the max-min formulation of the fuzzy sets obtained from the fuzzy goals, we calculate the fuzzy decision sets (Z and Z^*) that contain the feasible paths and the desirable paths, respectively, along with the degrees to which they are acceptable. To efficiently solve large instances of FOP, we also present a parallel algorithm on CREW PRAM model.

1. Introduction

The orienteering problem (OP) is an NP-hard problem, derived from the game of orienteering where the player is required to start from the initial control point and arrive at the final control point within the specified time limit, at the same time collecting the rewards (Score) assigned to each of the checkpoints that link the initial control point to the final control point. All those players who arrive at the final control point after the time expires are disqualified and the player reaching the final control point within the allotted time and with the maximum collected score is declared as the winner of the game. This game reflects many real life situations related to the field of logistics, tourism, building telecommunication networks, home delivery systems, and so forth, which can be modelled as OP [1]. There are four types of OP which include the simple orienteering problem, team orienteering problem, orienteering problem with time window, and team orienteering problem with time window. The OP can be observed as a combination of two well-known problems, that is, travelling salesman problem (TSP) and the knapsack problem (KP), where the objective of maximizing the score is derived from KP and the objective of minimizing

the time taken to travel from the initial control point and final control point is similar to the objective of TSP with the difference that in TSP all the vertices connecting the source to the target should be visited once, but in OP it is not necessary to visit all the intermediate checkpoints [1]. One real life application of OP is in the tourism industry where tourists come to visit the cities which are rich in culture and traditions and have some historical significance. The aim is to visit as many locations (including palaces, museums, monuments, and restaurants) as possible within their duration of stay. Due to the restriction of time it is not possible to visit all the places, so depending upon the choice and taste of the tourist, the guide can assign priorities to the various locations to be visited and then explore as many of them as possible within the time frame available so as to make their trip as economical and as beneficial as possible [2]. As seen in the stated example, the two parameters involved, that is, time and score, cannot be determined precisely as the priorities differ from one tourist to another and the time taken to travel from one location to another cannot be predicted exactly. So, the uncertainty in both the parameters can be modelled in a better way by representing them using fuzzy numbers instead of crisp or real numbers.

Since the orienteering problem is NP-hard, several heuristic algorithms and a few approximation algorithms are available in the literature. Some of the first heuristics for OP were the stochastic (S-algorithm) and deterministic (D-algorithm) ones suggested by Tsiligirides in 1984 [3]. In 1987, a center-of-gravity heuristic was proposed by Golden et al. [4]. A four-phase heuristic was introduced by Ramesh and Brown in 1991 [5]. An artificial neural network based approach was proposed by Wang et al. in 1995 [6] and, in 1996, a five-step heuristic was suggested by Chao et al. [7]. A branch-and-cut heuristic for OP was stated by Fischetti et al. in 1998 [8]. The first genetic algorithm for OP was developed by Tasgetiren in 2001 [9]. In 2002, Liang et al. proposed and compared a tabu search and an ant colony optimization approach for OP [10]. The approximations known for Prize Collecting Travelling Salesman Problem can be easily extended for the unrooted version of OP [11, 12] and, for the rooted version, a constant factor approximation was suggested by Blum et al. [13]. A heuristic method based on the greedy randomized adaptive search procedure and the path relinking methodologies was proposed in 2013 by Campos et al. [14].

This paper reports the first use of fuzzy logic to tackle the uncertainty involved in the two parameters (score and time). We introduce the fuzzy orienteering problem (FOP) and model the parameters involved as trapezoidal fuzzy numbers (TFNs). Several types of fuzzy numbers exist like Gaussian fuzzy numbers, exponential fuzzy numbers, quadratic fuzzy numbers, triangular fuzzy numbers, and trapezoidal fuzzy numbers, but here we prefer TFN because its computation is simple; it has a linear membership function and is the most generic class, so it is widely used in the applied engineering problems and in the scientific field.

In Section 2, we state necessary definitions and Section 3 provides the explanation of fuzzy optimization. In Section 4, the problem definition is stated. Section 5 explains the fuzzy formulation of OP along with the steps to deal with it and obtain a solution. The proposed method of solving FOP is demonstrated in Section 6 through an illustrative example. In Section 7, the parallel algorithm for FOP is presented and, finally, the paper is concluded in Section 8.

2. Preliminaries

Definition 1 (trapezoidal fuzzy numbers (TFNs)). A fuzzy number is an extension of the crisp or real number which refers to a set of possible values represented by a membership function, each of which is assigned a weight called its grade of membership between 0 and 1 whereas the crisp number is associated with a single value.

For any crisp set X which is a subset of the universal set U , the membership function is denoted by μ_X which assigns a value within $[0, 1]$ called the grade of membership to every element of X . So, the set $\tilde{X} = \{(x, \mu_{\tilde{X}}(x)) : x \in U\}$ is called a fuzzy set. A fuzzy set \tilde{X} , whose membership function possess the properties stated below, is called a fuzzy number [15]:

- (a) $\mu_{\tilde{X}}$ is piecewise continuous,
- (b) \tilde{X} is convex ($\mu_{\tilde{X}}(\alpha x_1 + (1 - \alpha)x_2) \geq \min(\mu_{\tilde{X}}(x_1), \mu_{\tilde{X}}(x_2)) \forall x_1, x_2 \in \mathbb{R}, \forall \alpha \in [0, 1]$),

- (c) \tilde{X} is normal ($\mu_{\tilde{X}}(x_0) = 1, x_0 \in \mathbb{R}$), where \mathbb{R} is the universal set of real numbers.

Fuzzy numbers are used to represent those parameters which are imprecise and, in several real life applications, parameters like cost, time, demand, capacity, and so forth are involved for which it is not possible to specify exact values. In these situations, it is appropriate to use a fuzzy number instead of a crisp number as they provide a more practical presentation of the real world. There are several types of fuzzy numbers, each of which can be defined by a membership function. In this paper, to represent the two parameters time and score as fuzzy numbers, we use TFN. For a trapezoidal fuzzy number represented as $\tilde{A} = (a_1, a_2, a_3, a_4)$, the membership function is as follows [15]:

$$\mu_{\tilde{A}}(x) = \begin{cases} 0, & x < a_1 \\ \frac{x - a_1}{a_2 - a_1}, & a_1 < x \leq a_2 \\ 1, & a_2 < x < a_3 \\ \frac{a_4 - x}{a_4 - a_3}, & a_3 \leq x < a_4 \\ 0, & x > a_4, \end{cases} \quad (1)$$

where $a_1 \leq a_2 \leq a_3 \leq a_4$.

Definition 2 (addition of TFN). In applications where the goal is to maximize or minimize some fuzzy quantity, some intermediate mathematical operations are to be performed which mostly include addition. For the problem stated here, we need to maximize the score acquired within a given time bound and for this we need to add the weights assigned to the vertices to calculate the score and the weights assigned to the edges to determine the time taken by a path. As stated the parameters here are considered to be TFNs and the mathematical operations performed on fuzzy numbers (like addition) differ from the usual crisp case.

Two TFNs represented as $\tilde{A} = (a_1, a_2, a_3, a_4)$ and $\tilde{B} = (b_1, b_2, b_3, b_4)$ can be added using the following formula [15]:

$$\begin{aligned} \tilde{A} + \tilde{B} &= (a_1, a_2, a_3, a_4) + (b_1, b_2, b_3, b_4) \\ &= (a_1 + b_1, a_2 + b_2, a_3 + b_3, a_4 + b_4). \end{aligned} \quad (2)$$

Definition 3 (ranking of TFN). In case of crisp numbers, there exists a natural ordering which clearly states that a real number is greater/smaller than other real numbers, but the absence of such a natural ordering in case of fuzzy numbers leads to the requirement of formulating a ranking method, using which the fuzzy numbers can be ordered. Here we use the circumcenter of centroids (COC) method for ranking TFN. Several methods are available in the literature for ranking of TFN, most of which use the centroid of the trapezoid for ranking considering it to be a balancing point, but in the method suggested by Rao and Shankar, the COC value is used for the ranking purpose. The COC is considered to be a better point of reference as it is obtained by calculating the circumcenter of the triangle formed by joining

the centroids of the three subparts of the trapezoid which include two triangles and one rectangle [16].

As suggested by Rao and Shankar [16], if $\tilde{A} = (a_1, a_2, a_3, a_4)$, then

$$\text{COC}(\tilde{A}) = (x, y) = \left(\frac{a_1 + 2a_2 + 2a_3 + a_4}{6}, \frac{(2a_1 + a_2 - 3a_3)(2a_4 + a_3 - 3a_2) + 5}{12} \right). \quad (3)$$

Using this value of $\text{COC}(\tilde{A})$, the rank of the TFN \tilde{A} is obtained using the following formula:

$$R(\tilde{A}) = \sqrt{x^2 + y^2}. \quad (4)$$

If we have two fuzzy numbers \tilde{A} and \tilde{B} , then $\tilde{A} > \tilde{B}$ if $R(\tilde{A}) > R(\tilde{B})$ and vice versa. When $R(\tilde{A}) = R(\tilde{B})$, it is difficult to determine the ordering and some additional information is used as suggested by Rao and Shankar to tackle the situation.

Index of Optimism. The degree of optimism of the decision maker was taken care of by this index as stated by Rao and Shankar in [16]. So the decision maker's view point was also considered and, for any TFN $\tilde{A} = (a_1, a_2, a_3, a_4)$ and $\text{COC}(\tilde{A})$, the formula for the index is as follows [16]:

$$I_\delta(\tilde{A}) = \delta y + (1 - \delta)x \quad \text{where } \delta \in [0, 1]. \quad (5)$$

Larger value of δ shows that the decision maker is optimistic, whereas smaller values depict the situation where the decision maker is pessimistic.

Index of Modality. Another index was introduced by Rao and Shankar to tackle the situation where the rank of two TFNs becomes equal because the index of optimism alone may not be sufficient in such a case to determine the ranking as it considers only the importance of extreme values. The index of modality considers the importance of central value as well as the extreme values and the formula to determine the index is as follows [16]:

$$I_{\delta, \gamma}(\tilde{A}) = \gamma \left(\frac{x + y}{2} \right) + (1 - \gamma) I_\delta(\tilde{A}), \quad (6)$$

where $\gamma \in [0, 1]$.

If we have two TFNs \tilde{A} and \tilde{B} , with $R(\tilde{A}) = R(\tilde{B})$, then $\tilde{A} > \tilde{B}$ when $I_{\delta, \gamma}(\tilde{A}) > I_{\delta, \gamma}(\tilde{B})$ and vice versa.

Definition 4 (expected value of trapezoidal fuzzy number (EV)). To determine the grade of membership of a TFN, that is, to find out the degree to which the TFN satisfies the specified requirement, we need to calculate its expected value and then use Definition 1. The formula for expected value as used by Jimenez et al. in [17] is as follows.

For a given TFN $\tilde{A} = (a_1, a_2, a_3, a_4)$,

$$\text{EV}(\tilde{A}) = \frac{1}{4} (a_1 + a_2 + a_3 + a_4). \quad (7)$$

This defuzzification technique is used as it is the simplest possible and gives satisfactory results for the fuzzy orienteering problem. It will be interesting to study other defuzzification techniques although the general conclusions of the paper will remain unaffected.

Definition 5 (fuzzy decision set (Z)). Fuzzy decision set is a set of elements providing a feasible solution to the stated problem. The fuzzy linear programming problem can have several goals, each represented by a membership function and a fuzzy set (F_i) containing the elements along with their grades of membership obtained using the membership function [17].

We define the fuzzy decision as

$$Z = F_1 \cap F_2 \cap \dots \cap F_l \quad (8)$$

$$\text{i.e. } \mu_Z(x) = \mu_{F_1}(x) * \mu_{F_2}(x) * \dots * \mu_{F_l}(x),$$

where $*$ represents a t -norm which can be any operation like *minimum*, *algebraic product*, and so forth. In case of OP, $*$ represents *minimum* operation. To obtain the most desirable solution, we determine the element with the highest membership degree in the fuzzy decision set Z , that is, the value of x that maximizes the membership function of fuzzy decision Z denoted by x^* , using the equation stated below:

$$\mu_{Z^*}(x^*) = \max \{ \mu_Z(x) \}, \quad (9)$$

where Z^* denotes the fuzzy set of the most desirable solutions.

3. Fuzzy Optimization

In engineering design or decision making problems, a large number of feasible solutions are available and to choose the solution which is the best one from this set, we need to concentrate on the uncertainty associated with the variables that lead to the optimal solution. Probabilistic concepts can take care of the randomness that arises due to natural fluctuation or natural variations, but the uncertainty that comes into existence due to qualitative statements, vague statements, vague nature of the objective, and linguistic statements showing the willingness of the decision maker (like the solution is acceptable, low, satisfactory, etc.) cannot be addressed through probabilistic concepts, so we introduce the concept of fuzzy logic in solving the optimization problems. In the crisp definition of optimization problems, we have crisp conditions where solutions violating the constraints or not satisfying the objective function are completely unacceptable, but, in fuzzy optimization, the concept of degree is introduced. The solution becomes a matter of degree; that is, degree of acceptability or degree of satisfaction is associated with the constraints and the objective functions, and this way we provide a latitude to the acceptability of a solution. This degree of acceptability linked with the objective functions and constraints can be reflected through fuzzy membership functions.

To deal with the situations where several stakeholders vaguely state their preferences as constraints or objective functions using linguistic statements, we convert these statements into fuzzy sets or fuzzy membership functions and

then using some technique find out the best “compromise solution.” In fuzzy optimization, we do not distinguish between the objective functions and the constraints; instead we refer to them as fuzzy goals, represented in the form of fuzzy sets defined by their respective membership functions. So, the latitude or uncertainty present in the decision making is tackled through these membership functions. In addition to the fuzzy goals, we can also have crisp constraints modelling the physical conditions or technological feasibility that have to be met in a particular solution.

The whole idea of fuzzy optimization is to allow for latitude in the constraints and flexibility in the objective function. Instead of a 0-1 type solution, we allow for some violation of the original constraints to some degree, set certain limit for the objective function, and accept solutions on both sides of the limit to different degrees. The objective function and the set of constraints are converted into fuzzy sets; their associated membership functions are defined and then all the membership functions are combined to determine the fuzzy decision. Through fuzzy optimization, the preferences of the decision maker are quantified and the uncertainty due to vagueness, imprecision, and so forth, which is common in decision making problems, is tackled using membership functions [18–20].

4. Problem Definition

The orienteering problem can be represented by a completely connected undirected graph $G(V, E)$, where $V = \{v_1, \dots, v_N\}$ is the set of vertices and E is the set of edges. A score S_i is associated with every vertex $v_i \in V$ and the time taken to traverse each edge $e_{ij} \in E$ is denoted by t_{ij} . The goal here is to determine a path P connecting any subset of V that necessarily includes the start vertex (v_1) and the end vertex (v_N), satisfies the time bound T_{\max} , and also maximizes the total collected score [1].

In this paper, we present the fuzzy orienteering problem (FOP) where the two quantities involved, that is, time and score, are considered to be fuzzy numbers. The reason to introduce fuzziness into the formulation of OP is that the crisp mathematical formulation is very strict in three ways: (1) the objective function should be either maximized or minimized, (2) none of the constraints should be violated as it leads to an infeasible solution, and (3) all constraints are given equal importance. However, these three necessary requirements lead to an unrealistic representation of the real world. By partly relaxing these using fuzzy logic, we can model the physical world in a more realistic manner. Several situations might exist in real life applications that can be easily represented in the fuzzy environment which may include the following [21].

- (1) The decision maker is not willing to maximize or minimize the objective function; instead he wants to reach some aspiration level like “improving the present fuel consumption situation to some extent” in transportation problems which cannot even be defined or stated in the crisp case.

- (2) Maybe the decision maker is willing to accept small violations in the constraints especially when the objective function deals with the aspiration levels and the less than or greater than relation is not required to be followed in the strict mathematical sense. Moreover, the parameters or variables involved may have vagueness which cannot be expressed in the crisp formulation.
- (3) In the crisp representation, all constraints are of equal importance but maybe, for the decision maker, different constraints have different importance and small violations of different constraints may be acceptable to different degrees.

In this fuzzy formulation of OP, we have recognized that the parameters time and score are fuzzy in nature and in the fuzzy version we provide latitude to the desired solution by relaxing the constraints to some extent and stating the degree up to which they are feasible. Here we do not distinguish between objective function and constraints; instead the two necessary conditions of maximizing the total collected score and following the time bound are represented as two goals which are conflicting as one is to be maximized and the other one is to be minimized. These are represented as linear membership functions and the rest of the constraints are considered to be crisp. The detailed explanation of the fuzzy formulation is presented in the next section.

5. Fuzzy Formulation of OP

The fuzzy version can be represented in the following way showing by tilde the parameters that have a fuzzy character:

$$\sum_{i=1}^{N-1} \sum_{j=2}^N \tilde{S}_{ij} x_{ij} \succeq S_{\min}, \quad (10)$$

$$\sum_{j=2}^N x_{1j} = 1, \quad (11)$$

$$\sum_{i=1}^{N-1} x_{iN} = 1, \quad (12)$$

$$\sum_{i=1}^{N-1} x_{ik} \leq 1 \quad \forall k = 2, \dots, N-1, \quad (13)$$

$$\sum_{j=2}^N x_{kj} \leq 1 \quad \forall k = 2, \dots, N-1, \quad (13)$$

$$\sum_{i=1}^{N-1} \sum_{j=2}^N \tilde{t}_{ij} x_{ij} \preceq T_{\max}, \quad (14)$$

$$2 \leq u_i \leq N \quad \forall i = 2, \dots, N, \quad (15)$$

$$u_i - u_j + 1 \leq (N-1)(1 - x_{ij}) \quad \forall i, j = 2, \dots, N, \quad (16)$$

$$x_{ij} \in \{0, 1\} \quad \forall i, j = 1, \dots, N. \quad (17)$$

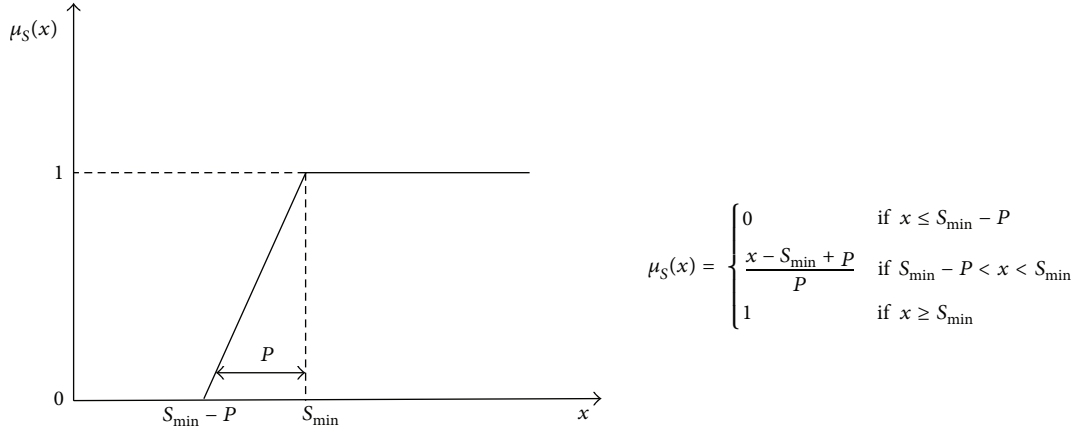


FIGURE 1: Membership function for total collected score of a path.

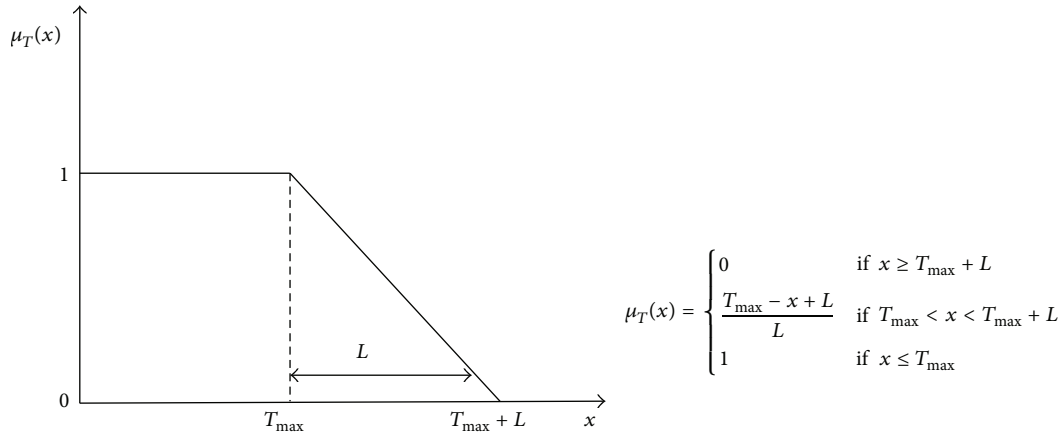


FIGURE 2: Membership function for total time taken to traverse a path.

As stated above, the constraints represented by (10) and (14) are considered to be fuzzy and represent the two conflicting goals of maximizing the total collected score and minimizing the time taken to traverse a path, respectively. The symbols \tilde{S}_i and \tilde{t}_{ij} in (10) and (14) represent the fuzzy score and fuzzy time values, respectively. The remaining constraints represented by (11), (12), (13), (15), (16), and (17) remain crisp. The variable u_i denotes the position of vertex v_i in the path and if vertex v_j is visited after vertex v_i , then $x_{ij} = 1$; else $x_{ij} = 0$. The necessary condition that each path starts in v_1 and ends in v_N is ensured by (11). The constraint that each path remains connected and no vertex is visited more than once in a path is taken care of by (12)-(13) and the requirement of eliminating subtours is implemented by (15)-(16) [1]. The symbols “ \succeq ” and “ \preceq ” are the fuzzy version of “ \geq ” and “ \leq ” representing the “fuzzy greater than or equal to” and “fuzzy less than or equal to,” respectively. The meaning of these symbols is that the constraints can be violated to some extent and, depending on the importance of the constraint, this violation leads to different degree of acceptance [22]. For example, in (10), the symbol “ \succeq ” indicates that the total collected score of a particular path should be greater than

S_{\min} (most desirable case), but those paths which have their total collected score slightly less than S_{\min} , that is, up to $S_{\min} - P$, are also acceptable but to a lesser degree. Similarly, in (14), the total time taken to traverse a path should be less than T_{\max} (most acceptable case), but paths having their total time greater than T_{\max} are also acceptable to different degrees up to the limit of $T_{\max} + L$ as signified by the symbol “ \preceq .” The two fuzzy goals can be represented by their membership functions as shown in Figures 1 and 2. The fuzzy decision sets Z and Z^* are depicted in Figure 3. It denotes the best “compromise solution” which is obtained by the following max-min formulation:

$$\begin{aligned} Z &= F_1 \cap F_2, \\ \mu_Z(x) &= \min \{ \mu_T(x), \mu_S(x) \}, \\ \mu_{Z^*}(x^*) &= \max \{ \mu_Z(x) \} \\ &= \max \{ \min \{ \mu_T(x), \mu_S(x) \} \}. \end{aligned} \quad (18)$$

The intersection of the two fuzzy sets (F_1 and F_2) is the minimum value of the two fuzzy sets for each x , which forms the fuzzy decision set Z (as shown by dark straight lines) and

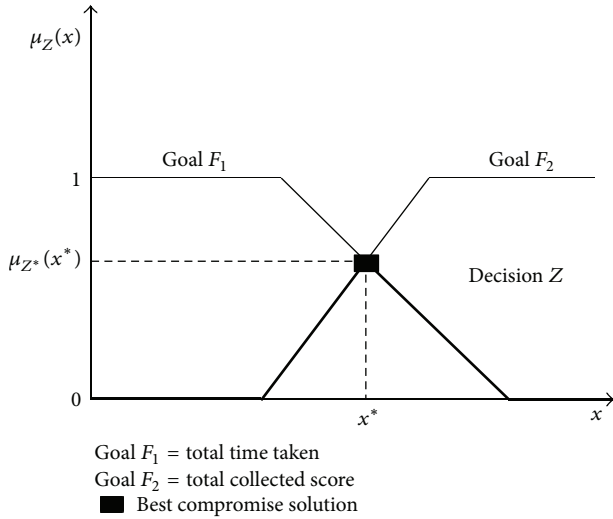


FIGURE 3: Fuzzy decision sets Z and Z^* .

the maximum value of this decision set Z forms the other set Z^* which holds the most desirable solution (as shown by the dashed line) of Figure 3.

FOP Algorithm. The following are the steps to determine the path that maximizes the total collected score within the specified time limit for a given $G(V, E)$ with N nodes.

- (1) Determine all the possible paths (W_m) connecting v_1 and v_N (using (11), (12), (13), (15), (16), and (17)).
- (2) For each possible path,
 - (a) calculate the total time taken to traverse the path and the total collected score (using Definition 2 and (2));
 - (b) calculate the expected value of the total time taken to traverse the path and the total collected score (using Definition 4 and (7));
 - (c) calculate its membership degree for the membership function of time (using (14) and Figure 2); let it be denoted by fuzzy set F_1 ;
 - (d) calculate its membership degree for the membership function of score (using (10) and Figure 1); let it be denoted by fuzzy set F_2 .
- (3) Determine the feasible paths denoted by the fuzzy decision set Z obtained using Definition 5 and (8).
- (4) The most desirable path (final solution) is denoted by the fuzzy decision set Z^* obtained using Definition 5 and (9).
- (5) If the fuzzy decision set Z^* contains more than one path, the total collected score (S_i) for each of the paths

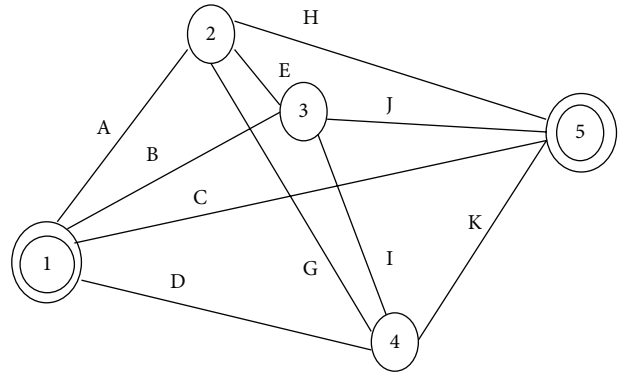


FIGURE 4: Input graph $G(V, E)$ with source vertex = 1 and destination vertex = 5.

TABLE 1: The values of edge weights (time taken to travel from one node to another).

Edge	Label	Fuzzy time values
e_{12}	A	(1, 4, 6, 9)
e_{13}	B	(6, 9, 13, 16)
e_{14}	D	(14, 15, 21, 22)
e_{15}	C	(4, 6, 8, 10)
e_{23}	E	(2, 3, 3, 4)
e_{24}	G	(5, 8, 8, 11)
e_{25}	H	(14, 18, 22, 26)
e_{34}	I	(5, 8, 12, 15)
e_{35}	J	(0, 2, 10, 12)
e_{45}	K	(1, 2, 2, 3)
		$T_{\max} = 20, L = 15$

TABLE 2: The values of node weights (score values).

Node	Label	Fuzzy score values
v_1	1	(1, 2, 8, 9)
v_2	2	(8, 9, 11, 12)
v_3	3	(3, 5, 9, 11)
v_4	4	(17, 20, 24, 27)
v_5	5	(1, 2, 4, 5)
		$S_{\min} = 25, P = 13$

in Z^* can be ranked (using Definition 3 and (3), (4), (5), and (6)) to determine the path that maximizes the total collected score.

6. Illustrative Example

Consider the network $G(V, E)$ in Figure 4 with total number of nodes $N = 5$ and the value for edge weights and node weights of G are stated in Tables 1 and 2, respectively.

Step 1. All possible paths connecting v_1 and v_5 are found that satisfy (11), (12), (13), (15), (16), and (17) and are as follows:

- W_1 : 1-5,
- W_2 : 1-2-5,
- W_3 : 1-3-5,
- W_4 : 1-4-5,
- W_5 : 1-2-3-5,
- W_6 : 1-3-2-5,
- W_7 : 1-2-4-5,
- W_8 : 1-4-2-5,
- W_9 : 1-3-4-5,
- W_{10} : 1-4-3-5,
- W_{11} : 1-2-3-4-5,
- W_{12} : 1-2-4-3-5,
- W_{13} : 1-3-4-2-5,
- W_{14} : 1-4-3-2-5,
- W_{15} : 1-4-2-3-5,
- W_{16} : 1-3-2-4-5.

Step 2. (a) Table 3 shows the total time taken to traverse and total collected score for each of the above stated paths which is calculated using Definition 2 and (2).

(b) Next we calculate the expected value of the total time taken to traverse the path and the total collected score using Definition 4 and (7) for each of the paths W_1 to W_{16} as shown in Table 4.

(c) and (d) The grade of membership of each possible path from W_1 to W_{16} is determined for both the membership functions (time and score) using (10) and (14) and Figures 1 and 2 and their values are shown in Table 5.

So, the two fuzzy sets F_1 and F_2 for the total time taken and total collected score, respectively, are as follows:

$$\begin{aligned}
 F_1 = \{ & W_1/1, W_2/0.67, W_3/1, W_4/1, W_5/1, W_6/0.06, \\
 & W_7/1, W_8/0, W_9/0.8, W_{10}/0.06, W_{11}/1, W_{12}/0.4, \\
 & W_{13}/0, W_{14}/0, W_{15}/0, P_{16}/0.73 \}, \\
 F_2 = \{ & W_1/0, W_2/0.23, W_3/0, W_4/1, W_5/0.77, \\
 & W_6/0.77, W_7/1, W_8/1, W_9/1, W_{10}/1, W_{11}/1, W_{12}/1, \\
 & W_{13}/1, W_{14}/1, W_{15}/1, W_{16}/1 \}.
 \end{aligned}
 \tag{20}$$

Step 3. Now the fuzzy decision set Z is obtained using Definition 5 and (8) to determine all the feasible paths along with the membership grades stating the degree up to which each of the feasible paths is acceptable.

TABLE 3: The total time taken and total collected score values for each of the paths.

Path	Total time taken to traverse	Total collected score
W_1	(4, 6, 8, 10)	(1, 2, 8, 9)
W_2	(15, 22, 28, 35)	(9, 11, 19, 21)
W_3	(6, 11, 23, 28)	(4, 7, 17, 20)
W_4	(15, 17, 23, 25)	(18, 22, 32, 36)
W_5	(3, 9, 19, 25)	(12, 16, 28, 32)
W_6	(22, 30, 38, 46)	(12, 16, 28, 32)
W_7	(7, 14, 16, 23)	(26, 31, 43, 48)
W_8	(33, 41, 51, 59)	(26, 31, 43, 48)
W_9	(12, 19, 27, 34)	(21, 27, 41, 47)
W_{10}	(19, 25, 43, 49)	(21, 27, 41, 47)
W_{11}	(9, 17, 23, 31)	(29, 36, 52, 59)
W_{12}	(11, 22, 36, 47)	(29, 36, 52, 59)
W_{13}	(30, 43, 55, 68)	(29, 36, 52, 59)
W_{14}	(35, 44, 58, 67)	(29, 36, 52, 59)
W_{15}	(21, 28, 42, 49)	(29, 36, 52, 59)
W_{16}	(14, 22, 26, 34)	(29, 36, 52, 59)

TABLE 4: The expected value of the total time taken to traverse the path and the total collected score for each of the paths.

Path	EV (total time taken)	EV (total collected score)
W_1	7	5
W_2	25	15
W_3	17	12
W_4	20	27
W_5	14	22
W_6	34	22
W_7	15	37
W_8	46	37
W_9	23	34
W_{10}	34	34
W_{11}	20	44
W_{12}	29	44
W_{13}	49	44
W_{14}	51	44
W_{15}	35	44
W_{16}	24	44

Here, $\mu_Z(x) = \min\{\mu_T(x), \mu_S(x)\}$, where $x = W_1$ to W_{16} is used to calculate Z :

$$\begin{aligned}
 Z = \{ & W_1/0, W_2/0.23, W_3/0, W_4/1, W_5/0.77, W_6/0.06, \\
 & W_7/1, W_8/0, W_9/0.8, W_{10}/0.06, W_{11}/1, W_{12}/0.4, \\
 & W_{13}/0, W_{14}/0, W_{15}/0, P_{16}/0.73 \}.
 \end{aligned}
 \tag{21}$$

Step 4. Finally, the fuzzy decision set Z^* is obtained using Definition 5 and (9) that contains the desirable path:

$$Z^* = \{W_4/1, W_7/1, W_{11}/1\}.
 \tag{22}$$

Step 5. As can be observed from Step 4, there are three paths that are desirable as they fulfil all the crisp constraints and

TABLE 5: The grade of membership of each possible path for the membership functions of both time and score.

Path (W_m)	$\mu_T(W_m)$	$\mu_S(W_m)$
W_1	1	0
W_2	0.67	0.23
W_3	1	0
W_4	1	1
W_5	1	0.77
W_6	0.06	0.77
W_7	1	1
W_8	0	1
W_9	0.8	1
W_{10}	0.06	1
W_{11}	1	1
W_{12}	0.4	1
W_{13}	0	1
W_{14}	0	1
W_{15}	0	1
W_{16}	0.73	1

the two fuzzy goals. To conclude with one most desirable path among the three available in Z^* , we use the ranking method

$$(a) \text{ speed up} = \frac{\text{runtime of the best known sequential algorithm}}{\text{runtime of the parallel algorithm for a } p\text{-processor machine}}$$

$$(b) \text{ total work done} = p * \text{runtime of the parallel algorithm for a } p\text{-processor machine} \quad (23)$$

$$(c) \text{ efficiency} = \frac{\text{runtime of the best known sequential algorithm}}{\text{total work done}}.$$

Any parallel algorithm is said to be work optimal if it has a linear speedup and an efficiency of 1.

If we have a graph with n nodes, then the maximum number of edges possible in the path connecting the source vertex (v_1) and the destination vertex (v_n) is $n - 1$, so the total number of possible paths can be calculated using the following formula:

$$\text{Total no. of paths } (p) = \sum_{i=1}^{n-1} \frac{(n-2)!}{[n-(i+1)]!}. \quad (24)$$

Figure 5 shows the several steps of the proposed parallel algorithm for FOP.

In the parallel formulation of FOP, the module SEQ-NOP computes the total number of possible paths (p) for a given input graph $\{G, v_1, v_n\}$ using (24) sequentially. We assume that the number of processors is equal to the total number of paths (p). This value is utilized by the following parallel modules to output the path in G connecting v_1 and v_n that maximizes the total collected score and minimizes the total travel time. If the module PAR-MAX returns more than one path with the same maximum value of membership value (MV), then the module SEQ-RDP sequentially ranks the total collected

described in Definition 3 and use (3), (4), (5), and (6) to rank the score of each path in Z^* and determine the best one. The values for which are shown in Table 6.

So, the most desirable path is W_{11} as it has the highest rank.

7. Parallel Formulation of FOP

To solve the FOP more efficiently and to yield a better and improved performance when applied on large instances, we present a parallel algorithm for FOP that uses the CREW PRAM (Concurrent Read Exclusive Write Parallel Random Access Machine) model. In a PRAM model, several processors are connected to a common block of shared memory. This shared memory can be accessed by all the processors and these processors communicate with each other by reading from or by writing to the shared memory. In a PRAM model, each processor is like a sequential RAM and can perform the arithmetic operations, assignment, comparison, memory access, and so forth in unit time. In CREW PRAM, more than one processor can read concurrently from the same cell of the shared memory but cannot write into the same cell concurrently; that is, write operation has to be exclusive [23, 24]. For each parallel algorithm, the following terms are computed to determine whether it is work optimal or not [24]:

score (TS) of the paths with the same maximum value of MV using the COC method of ranking fuzzy numbers and using Definition 3 and (3), (4), (5), and (6). Finally, the path with the highest rank is returned as the *most desirable path*.

The pseudocode for each of the modules is as follows.

PAR-PATH. See Pseudocode 1. The p valid paths can be computed in $O(1)$ time using p CREW PRAM processors.

So,

$$\text{speed up} = \frac{O(p)}{O(1)} = O(p),$$

$$\text{total work done} = O(1) * p = O(p), \quad (25)$$

$$\text{efficiency} = \frac{O(p)}{O(p)} = 1.$$

Therefore, this parallel module is work optimal.

PAR-TOT. See Pseudocode 2. The values for TTT and TS for each of the paths in D_k can be obtained in $O(1)$ time using p CREW PRAM processors.

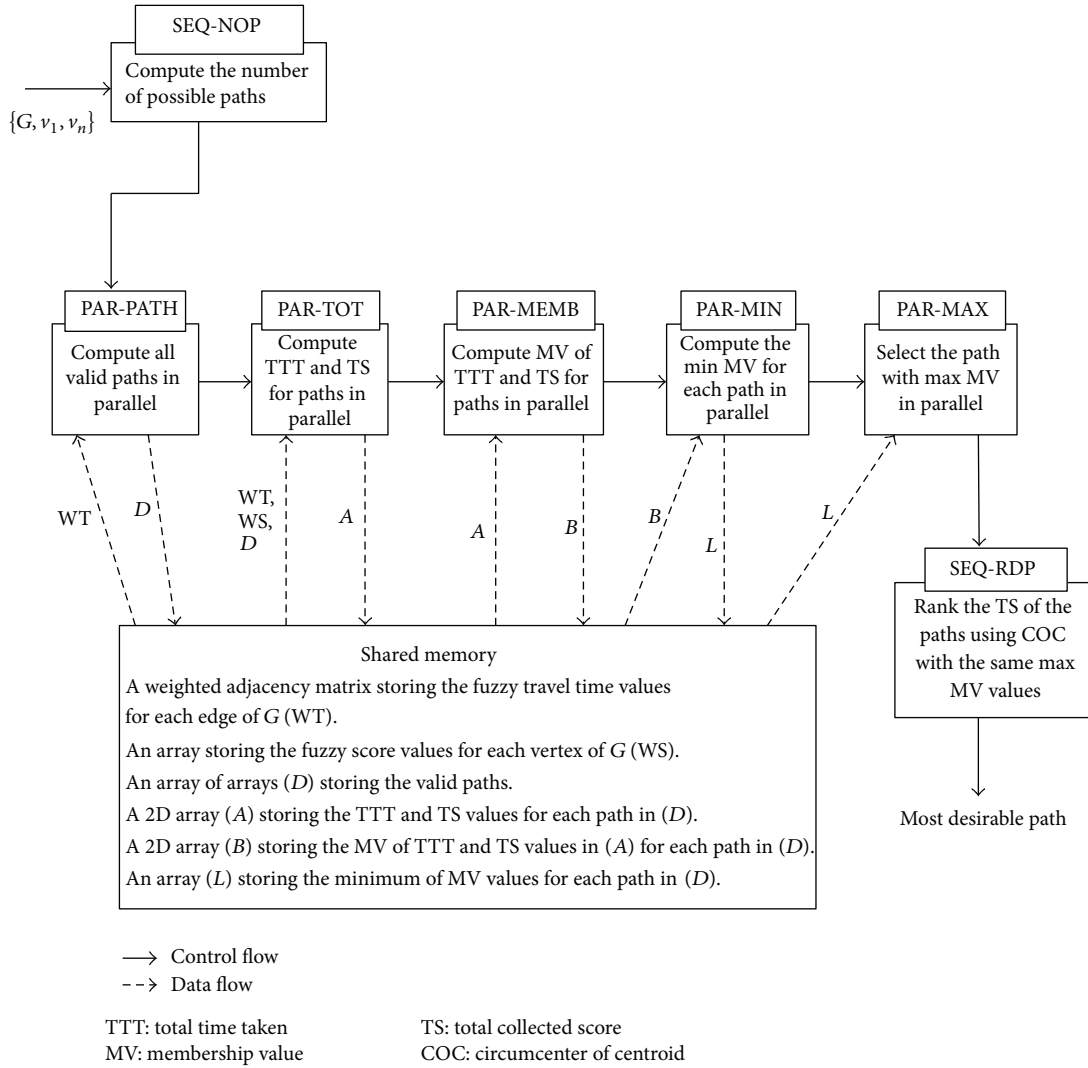


FIGURE 5: Several steps of the parallel formulation of FOP.

```

Processor  $k$  (in parallel for  $1 \leq k \leq p$ ) does:
{
   $cp :=$  compute a unique path connecting  $v_1$  and  $v_n$ ;
  if ( $cp == VALID$ ) then                      // A path is valid if it satisfies the constraints stated in (11), (12), (13), (15), (16) and (17).
     $D_k := cp$ ;
  else
    DISCARD  $cp$ ;
}
    
```

PSEUDOCODE 1

TABLE 6: The ranks of the desirable paths.

Path	Score	Rank
W_4	(18, 22, 32, 36)	3
W_7	(26, 31, 43, 48)	2
W_{11}	(29, 36, 52, 59)	1

So,

$$\begin{aligned}
 \text{speed up} &= \frac{O(p)}{O(1)} = O(p), \\
 \text{total work done} &= O(1) * p = O(p), \\
 \text{efficiency} &= \frac{O(p)}{O(p)} = 1.
 \end{aligned}
 \tag{26}$$

Therefore, this parallel module is work optimal.

```

Processor  $k$  (in parallel for  $1 \leq k \leq p$ ) does:
{
  for ( $j = 0$ ) do:
  {
     $A_{kj} :=$  Calculate the TTT for each VALID path in  $D_k$ ;
    // TTT (Total Time Taken) for each VALID path is calculated using Definition 2 and (2).
  }
  for ( $j = 1$ ) do:
  {
     $A_{kj} :=$  Calculate the TS for each VALID path in  $D_k$ ;
    // TS (Total Collected Score) for each VALID path is calculated using Definition 2 and (2).
  }
}

```

PSEUDOCODE 2

```

Processor  $k$  (in parallel for  $1 \leq k \leq p$ ) does:
{
  for ( $j = 0$ ) do:
  {
     $B_{kj} :=$  Calculate the MV for TTT in  $A_{kj}$  for each VALID path in  $D_k$ ;
    // MV (Membership Value) for TTT (Total Time Taken) is calculated using Definition 4 and (7), (14) and Figure 2.
  }
  for ( $j = 1$ ) do:
  {
     $B_{kj} :=$  Calculate the MV for TS in  $A_{kj}$  for each VALID path in  $D_k$ ;
    // MV (Membership Value) for TS (Total Collected Score) is calculated using Definition 4 and (7), (10) and Figure 1.
  }
}

```

PSEUDOCODE 3

PAR-MEMB. See Pseudocode 3. The membership values for TTT and TS for each of the paths in D_k can be calculated in $O(1)$ time using p CREW PRAM processors.

So,

$$\begin{aligned} \text{speed up} &= \frac{O(p)}{O(1)} = O(p), \\ \text{total work done} &= O(1) * p = O(p), \\ \text{efficiency} &= \frac{O(p)}{O(p)} = 1. \end{aligned} \quad (27)$$

Therefore, this parallel module is work optimal.

PAR-MIN. See Pseudocode 4. The minimum of the two membership values for TTT and TS for each of the paths in D_k can be computed in $O(1)$ time using p CREW PRAM processors.

So,

$$\begin{aligned} \text{speed up} &= \frac{O(p)}{O(1)} = O(p), \\ \text{total work done} &= O(1) * p = O(p), \\ \text{efficiency} &= \frac{O(p)}{O(p)} = 1. \end{aligned} \quad (28)$$

Therefore, this parallel module is work optimal.

PAR-MAX. See Pseudocode 5. The maximum value amongst the values present in the array L can be determined in $O(p)$ time using p CREW PRAM processors.

So,

$$\begin{aligned} \text{speed up} &= \frac{O(p^2)}{O(p)} = O(p), \\ \text{total work done} &= O(p) * p = O(p^2), \\ \text{efficiency} &= \frac{O(p^2)}{O(p^2)} = 1. \end{aligned} \quad (29)$$

Therefore, this parallel module is work optimal.

```

Processor  $k$  (in parallel for  $1 \leq k \leq p$ ) does:
{
  for ( $j = 0$ ) do:
  {
     $L[k] := \text{MIN}(B[k][j], B[k][j + 1])$ 
    \\ Compute the minimum of the two values using Definition 5 and (8).
  }
}

```

PSEUDOCODE 4

```

Processor  $k$  (in parallel for  $1 \leq k \leq p$ ) does:
{
   $G[k] := \text{GRADE of } L[k]$ ;
  // GRADE of any element  $L[k]$  in  $L$  is equal to one plus the number of elements in  $L$  smaller than  $L[k]$ .
  PRINT the path in  $D$  with the greatest GRADE in  $G$ .
}

```

PSEUDOCODE 5

8. Conclusion

In this paper, we deal with the orienteering problem which is an NP-hard problem and suggest a method to solve the fuzzy orienteering problem. As stated before, in these types of problems, the parameters involved like time and score are naturally imprecise, so representing them using real or crisp numbers leads to an unrealistic presentation of the real world. To provide a realistic model, fuzzy numbers can be used as they are capable of modelling the uncertainty present in the parameters and here we use trapezoidal fuzzy numbers to represent the values of the parameters involved (i.e., time and score). In the literature, the integer program formulation of OP has been stated. For these kinds of formulations, the objective function (either maximization or minimization) and the constraints are to be followed strictly in mathematical sense. However, this may not be the case in real life application where the decision maker may be willing to relax the constraints to some extent to generate a solution that can achieve a level of satisfaction which cannot be represented by the crisp formulation, but the fuzzy version can take care of these requirements. In our work, the two necessary conditions of maximizing the total collected score and that too within the specified time bound is represented by two fuzzy goals and the other requirements like each vertex is visited once, no sub tours are formed etc. are represented as crisp constraints. By representing the two necessary conditions in the form of fuzzy goals we provide a latitude to the desired solution by relaxing the constraints as can be observed in Figures 1 and 2 where we have fixed a time bound T_{\max} and a minimum score S_{\min} for the desired solution, but we accept solution up to $T_{\max} + L$ and $S_{\min} - P$ up to different degrees and this helps the decision maker to achieve the level he wants in his solution. This approach requires that the shapes and parameters for the membership functions be decided depending on the application at hand

taking into account the extent of uncertainty prevailing. To make the proposed method applicable to large instances, we also present a work optimal parallel formulation of FOP. When provided with several graphs as input, further speedup can be expected since the stages will form a 5-stage linear pipeline providing a speedup of 5 ideally when all the stages are busy. In this way the method suggested above provides a more practical illustration of the physical world and generates a solution that is more appropriate and realistic.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

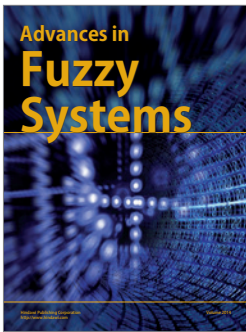
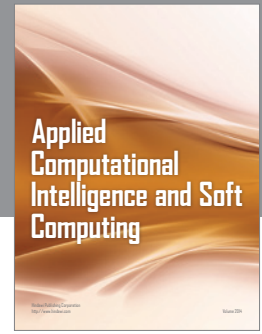
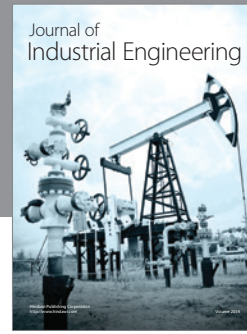
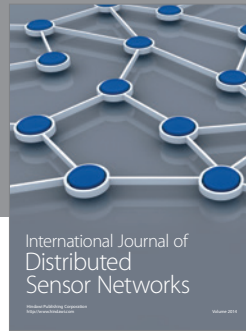
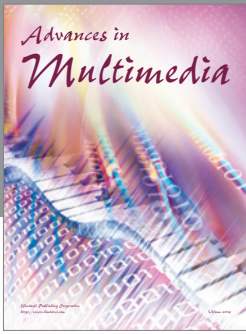
Acknowledgment

The first author would like to acknowledge the financial support by IIT (BHU) in terms of teaching assistantship.

References

- [1] P. Vansteenwegen, W. Souffriau, and D. van Oudheusden, "The orienteering problem: a survey," *European Journal of Operational Research*, vol. 209, no. 1, pp. 1–10, 2011.
- [2] M. Schilde, K. F. Doerner, R. F. Hartl, and G. Kiechle, "Metaheuristics for the bi-objective orienteering problem," *Swarm Intelligence*, vol. 3, no. 3, pp. 179–201, 2009.
- [3] T. Tsiligirides, "Heuristic methods applied to orienteering," *Journal of the Operational Research Society*, vol. 35, no. 9, pp. 797–809, 1984.
- [4] B. L. Golden, L. Levy, and R. Vohra, "The orienteering problem," *Naval Research Logistics*, vol. 34, no. 3, pp. 307–318, 1987.
- [5] R. Ramesh and K. M. Brown, "An efficient four-phase heuristic for the generalized orienteering problem," *Computers & Operations Research*, vol. 18, no. 2, pp. 151–165, 1991.

- [6] Q. Wang, X. Sun, B. L. Golden, and J. Jia, "Using artificial neural networks to solve the orienteering problem," *Annals of Operations Research*, vol. 61, no. 1, pp. 111–120, 1995.
- [7] I. Chao, B. Golden, and E. Wasil, "Theory and methodology—a fast and effective heuristic for the orienteering problem," *European Journal of Operational Research*, vol. 88, pp. 475–489, 1996.
- [8] M. Fischetti, J. J. S. González, and P. Toth, "Solving the orienteering problem through branch-and-cut," *INFORMS Journal on Computing*, vol. 10, no. 2, pp. 133–148, 1998.
- [9] M. Tasgetiren, "A genetic algorithm with an adaptive penalty function for the orienteering problem," *Journal of Economic and Social Research*, vol. 4, pp. 1–26, 2001.
- [10] Y.-C. Liang, S. Kulturel-Konak, and A. E. Smith, "Meta heuristics for the orienteering problem," in *Proceedings of the Congress on Evolutionary Computation (CEC '02)*, pp. 384–389, Honolulu, Hawaii, USA, May 2002.
- [11] B. Awerbuch, Y. Azar, A. Blum, and S. Vempala, "Improved approximation guarantees for minimum-weight k-trees and prize-collecting salesmen," *SIAM Journal on Computing*, vol. 28, pp. 254–262, 1999.
- [12] D. Johnson, M. Minkoff, and S. Phillips, "The prize collecting steiner tree problem: theory and practice," in *Proceedings of the 11th Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 760–769, 2000.
- [13] A. Blum, S. Chawla, D. R. Karger, T. Lane, A. Meyerson, and M. Minkoff, "Approximation algorithms for orienteering and discounted-reward TSP," in *Proceedings 44th Annual IEEE Symposium on Foundations of Computer Science (FOCS '03)*, pp. 1–10, October 2003.
- [14] V. Campos, R. Martí, J. Sánchez-Oro, and A. Duarte, "GRASP with path relinking for the orienteering problem," *Journal of the Operational Research Society*, vol. 65, no. 12, pp. 1800–1813, 2013.
- [15] A. Bansal, "Trapezoidal fuzzy numbers (a, b, c, d): arithmetic behavior," *International Journal of Physical and Mathematical Sciences*, vol. 2, no. 1, pp. 39–44, 2011.
- [16] P. P. Rao and N. R. Shankar, "Ranking fuzzy numbers with a distance method using circumcenter of centroids and an index of modality," *Advances in Fuzzy Systems*, vol. 2011, Article ID 178308, 7 pages, 2011.
- [17] M. Jimenez, M. Arenas, A. Bilbao, and M. V. Rodriguez, "Linear programming with fuzzy parameters: an interactive method resolution," *European Journal of Operational Research*, vol. 177, no. 3, pp. 1599–1609, 2007.
- [18] U. Kaymak and J. M. Sousa, *Weighted Constraints in Fuzzy Optimization*, Report Series Research in Management, ERIM Research Program: "Business Processes, Logistics and Information Systems", 2001, <http://repub.eur.nl/res/pub/85/erimrs20010403174009.pdf>.
- [19] J. Ramik, *Soft Computing: Overview and Recent Developments in Fuzzy Optimization*, 2001, http://irafm.osu.cz/research_report/118_softco01.pdf.
- [20] D. P. Loucks and E. V. Beek, *Water Resources Systems Planning and Management: An Introduction to Methods, Models and Applications*, UNESCO Publishing, 2005.
- [21] H. J. Zimmermann, "Fuzzy set theory," *WIREs Computational Statistics*, vol. 2, pp. 317–332, 2010.
- [22] F. Jarray, "Discrete tomography and fuzzy integer programming," *Iranian Journal of Fuzzy Systems*, vol. 8, pp. 41–48, 2011.
- [23] G. E. Blelloch and B. M. Maggs, "Parallel Algorithms," <http://homes.cs.washington.edu/~arvind/cs424/readings/BlellochM96b.pdf>.
- [24] E. Horowitz, S. Sahni, and S. Rajasekaran, *Computer Algorithms/C++*, Galgotia Publications, New Delhi, India, 1999.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

