CHAPTER 3

Intuitionistic Fuzzy Numbers and Intuitionistic Fuzzy Point

Abstract

This chapter is based on the concept of intuitionistic fuzzy sets introduced by Atanassov. A new type of intuitionistic fuzzy number called the Quasi-Gaussian intuitionistic fuzzy number has been proposed. As the intuitionistic fuzzy numbers lack the property of natural ordering, few ranking methods have been introduced. A centroid based ranking method for trapezoidal intuitionistic fuzzy number has been stated here which uses an eight variable representation and another ranking method called the Centroid of Centroids has been proposed. Similarly, for the Quasi-Gaussian fuzzy number a centroid based ranking method is introduced. The application of these fuzzy numbers and their ranking methods have been presented and experimentally verified for the constrained intuitionistic fuzzy shortest path problem. Also, the orienteering problem which is an NP-Hard problem has been considered and stated as a fuzzy integer program with fuzzy goals and crisp constraints and a fuzzy optimization technique for solving the intuitionistic fuzzy orienteering problem has been introduced that considers the hesitancy, aspiration levels etc. of the decision maker and provides some latitude to the solution process. Also, an intuitionistic fuzzy metric space has been proposed using the concept of intuitionistic fuzzy point and intuitionistic fuzzy scalars and the distance measure thus proposed has been applied on the orienteering problem.

In section 3.1, the concept of intuitionistic fuzzy sets is described. Section 3.2 presents the definition and application of trapezoidal intuitionistic fuzzy number and Quasi-Gaussian fuzzy number. The max-min formulation of intuitionistic fuzzy orienteering problem has been presented in section 3.3. The

concept of intuitionistic fuzzy metric space using intuitionistic fuzzy point and intuitionistic fuzzy scalars has been stated in section 3.4. Finally, the chapter is concluded in section 3.5.

3.1 Introduction

The idea of intuitionistic fuzzy sets (IFS) which is a generalization of fuzzy sets was proposed by Atanassov (1983). Though IFS is an extension of fuzzy sets proposed by Zadeh, there are several situations that can be modelled using IFS but cannot be represented using ordinary fuzzy sets. For example, let us consider a travelling salesman who has a limit on distance travelled that prevents him from visiting all the cities but he has some information about the cities where maximum sales can take place. Therefore, the aim here is to maximize the total sales within the limit on the distance travelled. If E is a set consisting of all those cities which can be visited by the salesman and $m \in E$ represents a city visited by the salesman, then the membership degree of the cities visited by the salesman can be stated as $\mu(m)$. In this case, an ordinary fuzzy set can be used but this ordinary fuzzy set cannot represent the situation where the need is to evaluate the number of cities that could not be visited by the salesman. For this, we need IFS where the degree of non-membership can be computed as $\nu(m) = 1 - \mu(m)$. Moreover, there can be a situation where the salesman visited a city but could not sell his product due to unavailability of the customer or shop being closed, then such situations can be represented through the hesitancy degree as $\pi(m) = 1 - \left(\mu(m) + \nu(m)\right)$. Also, there are a few operators like the modal operators that can be defined for IFS but not for ordinary fuzzy sets. These operators provide for a detailed estimation of the available information. Also, IFS is more powerful as it allows geometrical interpretation and helps in managing the existing uncertainty in a much better way (Atanassov, 2003). In IFS, two values are associated with every element of the set, one depicting the degree of belongingness and the other being the degree of non-belongingness. Both these values lie within the real unit interval $[0,1]$ (Ye, 2011).

In this chapter, we consider two types of intuitionistic fuzzy numbers (IFN) viz. trapezoidal intuitionistic fuzzy number (TIFN) and Quasi-Gaussian intuitionistic fuzzy number (QGIFN). In applications like constrained shortest path problems (CSPP) which can be applied in several real life situations, the major obstacle is to tackle the uncertainty that comes into play due to the parameters involved like cost, delay, time, energy etc. and at the same time provide the assurance of quality of service (QoS). The best way to deal with this imprecise nature of these parameters present in the network is to use fuzzy numbers. Here, we present a model where the parameters are represented as IFN and hence provide a more realistic picture of the practical situation as the uncertainty can be analyzed in a better way using the degrees of belongingness and non-belongingness. One problem that exists in the case of fuzzy representation also applies to IFN i.e., ranking of IFN as they also cannot be ordered naturally. Therefore, a centroid based ranking method for trapezoidal intuitionistic fuzzy number and Quasi-Gaussian intuitionistic fuzzy number has been proposed.

Also we consider the orienteering problem which is an NP-Hard problem. Several heuristics and approximation algorithms have been proposed to tackle the problem but here, for the first time the intuitionistic fuzzy version of the problem has been formulated and solved using the max-min formulation and a work-depth analysis of its parallel formulation has also been presented. The concept of intuitionistic fuzzy point (IFP) was introduced by Coker and Demirci (1995). The idea of IFP was then used by researchers to prove some relations and theorems (Akram, 2012; Sardar, Mandal, & Majumder, 2011). The concept of IFP can be used to study some general structures like those introduced in (Bustince, Barrenechea, & Pagola, 2006; Bustince, Barrenechea, & Pagola, 2008). In this chapter, we use the concept of IFP and intuitionistic fuzzy scalars for proposing a new definition of the intuitionistic fuzzy metric space. The proposed definition of distance metric is then applied to the orienteering problem.

3.2 Intuitionistic Fuzzy Numbers (IFN)

For an IFS, if the real line is the universe of discourse i.e., $U = \mathcal{R}$, then it is termed as an intuitionistic fuzzy number (IFN). An IFN can be represented as $Z = \{\langle x, \mu_Z(x), \nu_Z(x) \rangle : x \in \mathcal{R}\}$ and has the following properties (Grzegorzewski, 2003):

(1) The membership function and the non-membership function is fuzzy convex and fuzzy concave respectively.

(2) $\mu_Z(x_1) = 1$ and $\nu_Z(x_2) = 1$ for at least two points x_1 and x_2 belonging in U.

(3) The membership function (μ_X) and the non-membership function (ν_X) is upper semi-continuous and lower semi-continuous respectively.

3.2.1 Trapezoidal Intuitionistic Fuzzy Number (TIFN)

The definition of a trapezoidal intuitionistic fuzzy number is presented below:

Let $A = \langle (a, b, c, d), (e, f, g, h) \rangle$ be a TIFN $where\ a, b, c, d, e, f, g, h \in \mathcal{R}$ such that $e \le a \le f \le b \le c \le g \le d \le h$ and the functions $L_A, M_A, N_A, K_A : \mathcal{R} \to [0, 1]$, then (Ye, 2011)

$$\mu_A(x) = \begin{cases} 0 & if\ x < a \\ L_A(x) & if\ a \le x < b \\ 1 & if\ b \le x \le c \\ M_A(x) & if\ c < x \le d \\ 0 & if\ d < x \end{cases} \tag{3.1a}$$

$$\nu_A(x) = \begin{cases} 1 & if\ x < e \\ N_A(x) & if\ e \le x < f \\ 0 & if\ f \le x \le g \\ K_A(x) & if\ g < x \le h \\ 1 & if\ h < x \end{cases} \tag{3.1b}$$

Where $L_A(x) = \frac{x-a}{b-a}$, $M_A(x) = \frac{x-d}{c-d}$, $N_A(x) = \frac{x-f}{e-f}$, $K_A(x) = \frac{x-g}{h-g}$

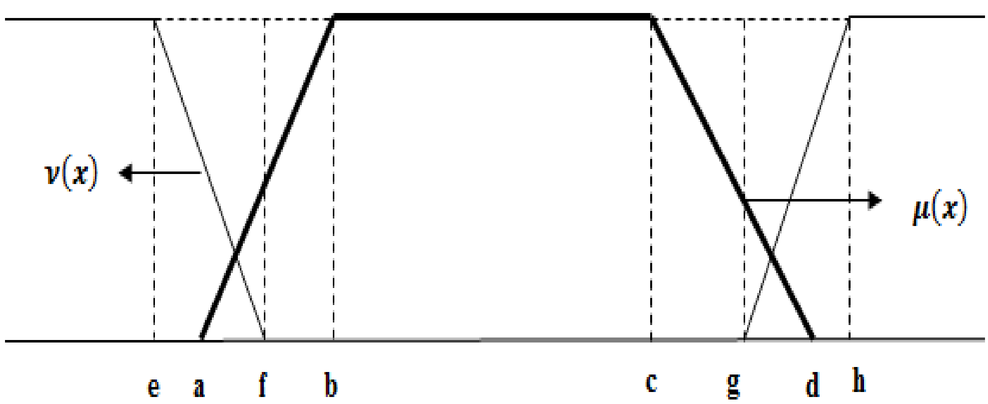


Fig. 3.1: Trapezoidal intuitionistic fuzzy number (TIFN)

(I) Addition of two TIFNs

In problems like constrained intuitionistic fuzzy shortest path problem (CIFSPP), the two important mathematical operations involved are addition and ranking for determining the delay constrained least cost path present in the network. Ranking of TIFNs has been discussed in detail in the next section. Here we state the equation for addition of TIFNs (Ye, 2011).

Let

$A_1 = \langle (a_1, b_1, c_1, d_1), (e_1, f_1, g_1, h_1) \rangle$ *and*

$A_2 = \langle (a_2, b_2, c_2, d_2), (e_2, f_2, g_2, h_2) \rangle$

be two TIFN, then

$$A_1 + A_2 = \langle (a_1, b_1, c_1, d_1), (e_1, f_1, g_1, h_1) \rangle + \langle (a_2, b_2, c_2, d_2), (e_2, f_2, g_2, h_2) \rangle$$

$$= \left\langle \begin{matrix} (a_1 + a_2, b_1 + b_2, c_1 + c_2, d_1 + d_2), \\ (e_1 + e_2, f_1 + f_2, g_1 + g_2, h_1 + h_2) \end{matrix} \right\rangle \tag{3.2}$$

(II) Ranking of TIFNs

(a) Centroid Method

To rank the TIFNs we introduce a centroid method of ranking in this chapter. Several other methods of ranking are also available in the literature and in (Varghese & Kuriakose, 2012), authors suggested a technique to compute the centroid of an IFN. However, they use a six parameter representation for an IFN and here we represent the TIFN using eight parameters as shown in Eqs. 3.1a, 3.1b and Fig. 3.1, taking into consideration the hesitancy involved and providing a better modelling of the practical situation.

To determine the centroid of a TIFN, let us consider a new fuzzy number $z: \mathbb{R} \rightarrow [0,1]$ such that

$$z(x) = \frac{(\mu - \nu)(x) + 1}{2} \tag{3.3}$$

where μ and ν represent the membership function and non-membership function respectively.

Now, the centroid $(C) = \frac{\int_e^h z(x)x\,dx}{\int_e^h z(x)dx}$ (3.4)

Using (3.3), we can say that

$$(C) = \frac{\int_e^h \frac{(\mu-\nu)(x)+1}{2} x\,dx}{\int_e^h \frac{(\mu-\nu)(x)+1}{2} dx} \tag{3.5}$$

Integrating the membership and non-membership functions separately, we get:

From (3.1a) and (3.1b), we can say

$$\int_e^h \mu(x)\,x\,dx = \int_a^b \left(\frac{x-a}{b-a}\right) x\,dx + \int_b^c (1)\,x\,dx + \int_c^d \left(\frac{x-d}{c-d}\right) x\,dx$$

$$= \frac{(c^2+d^2+cd)-(a^2+b^2+ab)}{6} \tag{3.6}$$

Similarly on integrating $\int_e^h [1-\nu(x)]\,x\,dx$, we get

$$\int_e^h [1-\nu(x)]\,x\,dx = \frac{(g^2+h^2+gh)-(e^2+f^2+ef)}{6} \tag{3.7}$$

Therefore, $\int_e^h z(x)x\,dx = \frac{(c^2+d^2+cd)+(g^2+h^2+gh)-(a^2+b^2+ab)-(e^2+f^2+ef)}{12}$ (3.8)

Similarly using (3.1a), (3.1b) and integration, we evaluate the function in the denominator of (3.4) and get the following equations:

$$\int_e^h \mu(x)\,dx = \frac{(c+d)+(a+b)}{2} \tag{3.9}$$

$$\int_e^h [1-\nu(x)]\,dx = \frac{(g+h)+(e+f)}{2} \tag{3.10}$$

Therefore,

$$\int_e^h z(x)\,dx = \frac{(c+d+g+h)-(a+b+e+f)}{4} \tag{3.11}$$

The formula for calculating the centroid of a TIFN is derived using (3.4), (3.8) and (3.11) and is presented below:

$$(C) = \frac{[(c^2+d^2+cd)+(g^2+h^2+gh)-(a^2+b^2+ab)-(e^2+f^2+ef)]}{3[(c+d+g+h)-(a+b+e+f)]} \tag{3.12}$$

(b) Centroid of Centroids Method (CoC)

To rank the TIFN, we introduce a technique called Centroid of Centroids (CoC). The centroid of a fuzzy number signifies its geometric centre and is denoted using the formula: $\int_{-\infty}^{\infty} xf(x)dx / \int_{-\infty}^{\infty} f(x)dx$. A trapezoid can be divided into three figures (two triangles and a rectangle) and finding out the centroid of each and joining them forms a triangle. The centroid of this resultant triangle can be considered to be a balancing point and a better point of reference. As the task here is to rank a trapezoidal intuitionistic fuzzy number, we evaluate the centroid for both the trapezoids using the following formula and Fig. 3.2.

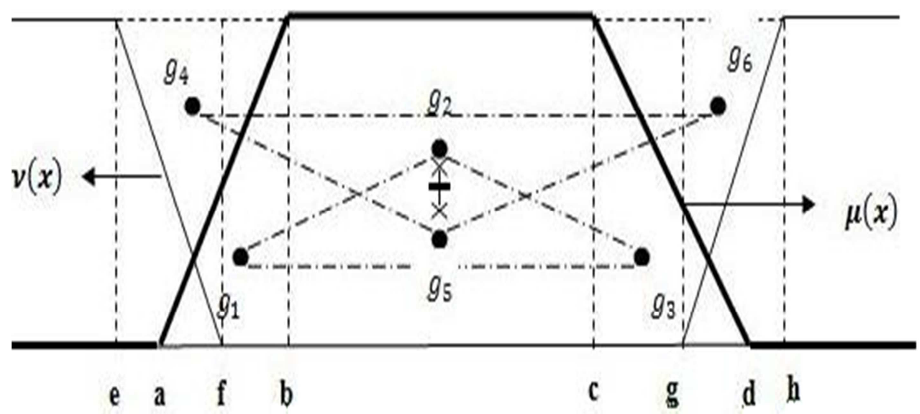


Fig. 3.2: The point of reference used for ranking a TIFN

The centroid of the triangle $g_1g_2g_3 = C_1$

$$C_1 = (x_1, y_1) = \left[\frac{(2a+b+7c+2d)}{18}, \frac{7}{18}\right] \tag{3.13}$$

The centroid of the triangle $g_4g_5g_6 = C_2$

$$C_2 = (x_2, y_2) = \left[\frac{(2e+f+2h+7g)}{18}, \frac{11}{18}\right] \tag{3.14}$$

Then, the value that can be used for ranking can be evaluated using the following formula:

$$Centroid\ of\ centroids\ (CoC) = \sqrt{\left(\frac{x_1+x_2}{2}\right)^2 + \left(\frac{y_1+y_2}{2}\right)^2} \tag{3.15}$$

After the CoC values are computed for the parameter involved i.e., the total cost or the total collected score of each feasible path, then $Rank\ (R)$ is assigned to each of the paths i.e., the path with higher CoC value gets a higher rank.

(III) Path Delay Discretization

As stated in chapter 2, CSPP is an NP-Complete problem and the technique suggested by Chen *et al.* (2008) to reduce it to a polynomial time solvable one is discretization. They proposed two types of discretization, namely randomized discretization and path delay discretization with the aim to provide better and accurate network functions and best utilization of limited resources. However, the effectiveness of this method depends upon the amount of error induced as a result of discretization.

For CIFSPP, we prefer path delay discretization algorithm (PDA) over randomized discretization because the problem of error accumulation is absent in case of PDA as it uses the method of interval partitioning for discretization of path delays (Chen, Song, & Sahni, 2008). The discretized delay for any path can be computed using the following equation:

$$d'(P) = \left\lfloor \frac{d(P)}{r} \lambda \right\rfloor \tag{3.16}$$

Where $\lfloor X \rfloor = floor(X)$ is the largest integer not greater than X.

Here, $d(P)$, r and λ signifies the total delay consumption value of a path, the delay constraint and the integer that bounds the delay constraint respectively.

(IV) Problem Definition

Networks can be modelled using graphs and can be denoted by the ordered pair $G(V, E)$ where V and E represents the set of nodes and edges respectively such that $|V| = n\ and\ |E| = m$. The cost and delay consumption value of each edge can be stated as $c(u,v)$ and $d(u,v)$ where $(u,v) \epsilon E$ symbolizes each edge of the graph G. To compute the total cost and total delay consumption value of each path represented as $c(P)$ and $d(P)$ respectively, the following equations can be used (Chen, Song, & Sahni, 2008):

$$d(P) = \sum_{(u,v)\in P} d(u,v) \tag{3.17}$$

$$c(P) = \sum_{(u,v)\in P} c(u,v) \tag{3.18}$$

A path P, connecting the specified source (s) and target (t) is said to be feasible if it satisfies the condition $d(P) \leq r$, and is called the cheapest feasible path if it obeys the constraint i.e., $d(P) \leq r$ and fulfils the objective of achieving the minimum cost $c(P_{s,t})$ amongst the available paths linking the origin and the destination (Chen, Song, & Sahni, 2008). As we are considering the intuitionistic fuzzy version of the problem CSPP, the cost of each edge is represented as either TIFN or QGIFN and to compute the cost of the path, the formula for addition of TIFN (Eq. 3.2) or addition of QGIFN (Eq. 3.23) is used depending upon the application. Also, to conclude with the cheapest feasible path, any of the ranking method suggested in section 3.2.1 (II) for TIFN and 3.2.2 (II) for QGIFN can be used.

(V) Application

(a) CIFSPP Algorithm (using TIFN)

$\boldsymbol{Get\_abcd(\ alpha1, stretch1, cost)}$

1. $a = cost - stretch1$
2. $b = a + alpha1$
3. $d = cost + stretch1$
4. $c = d - alpha1$

$\boldsymbol{Get\_efgh(\ alpha2, stretch2, cost)}$

1. $e = cost - stretch2$
2. $f = e + alpha2$
3. $h = cost + stretch2$
4. $g = h - alpha2$

$\boldsymbol{Initialize}\ (\boldsymbol{V}, \boldsymbol{s}, \boldsymbol{\lambda})$

1\. $\boldsymbol{for}$ $each\ vertex$ $\boldsymbol{v} \in \boldsymbol{V}$, $each$ $\boldsymbol{i} \in [\boldsymbol{0}, \dots \dots, \boldsymbol{\lambda}]$

2\. $\boldsymbol{Get\_abcd}(\ \boldsymbol{alpha1}, \boldsymbol{stretch1}, \boldsymbol{cost})$,

$\boldsymbol{Get\_efgh}(\ \boldsymbol{alpha2}, \boldsymbol{stretch2}, \boldsymbol{cost})$

3\. $w[v,i] \coloneqq\ \infty, \pi[v,i] \coloneqq NIL, z[v,i] \coloneqq\ \infty$

4\. $w[s,0] \coloneqq 0, z[s,i] \coloneqq 0$

5\. $\boldsymbol{end\ for}$

$\boldsymbol{Relax\_FPDA}(\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{i}, \boldsymbol{\lambda})$

1\. $i' \coloneqq\ floor\left(\frac{z[u,i] +\ d(u,v)}{r} \lambda\right)$

2\. $\boldsymbol{Get\_abcd}(\ \boldsymbol{alpha1}, \boldsymbol{stretch1}, \boldsymbol{cost})$,

$\boldsymbol{Get\_efgh}(\ \boldsymbol{alpha2}, \boldsymbol{stretch2}, \boldsymbol{cost})$

3\. $\boldsymbol{if}$ $i' \leq\ \lambda\ and\ w[v,i'] > w[u,i] + c(u,v)$

// Compare using Eq. 3.12 of section 3.2.1 (II) (a).

4\. $w[v,i'] \coloneqq w[u,i] + c(u,v)$

5\. $\pi[v,i'] \coloneqq u$

6\. $z[v,i'] \coloneqq min\{z[v,i'],\ z[u,i] + d(u,v)\}$

7\. $\boldsymbol{end\ if}$

$\boldsymbol{FPDA\_Dijkstra}(\boldsymbol{G}, \boldsymbol{s}, \boldsymbol{\lambda})$

1\. $\boldsymbol{Initialize}\ (\boldsymbol{V}, \boldsymbol{s}, \boldsymbol{\lambda})$

2\. $\boldsymbol{for}$ i = 0 to λ

3\. $Q \coloneqq V$

4\. $\boldsymbol{while}$ $Q \neq \varphi$

5\. $u \coloneqq Extract\_Min(Q)$

6\. $\boldsymbol{if}$ $w[u,i] = \infty$

7\. $\boldsymbol{break}$ $out\ of\ the$ $\boldsymbol{while}$ $loop$

8\. $\boldsymbol{end\ if}$

9\. $Q \coloneqq Q - \{\ u\ \}$

10\. $\boldsymbol{for}$ $every\ adjacent\ node$ $\boldsymbol{v}$ of $\boldsymbol{u}$

11\. $\boldsymbol{Relax\_FPDA}(\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{i}, \boldsymbol{\lambda})$

12\. $\boldsymbol{end\ for}$

13\. $\boldsymbol{end\ while}$

14\. $\boldsymbol{end\ for}$

$\boldsymbol{FPDA}(\boldsymbol{G}, \boldsymbol{s})$

1. $\lambda \coloneqq \lambda_0$
2. $\boldsymbol{do}$
3. $\lambda \coloneqq 2\lambda$
4. $\boldsymbol{FPDA\_Dijkstra}(\boldsymbol{G}, \boldsymbol{s}, \boldsymbol{\lambda})$
5. $\boldsymbol{while}\ \exists v \in d(P^v) > (1+\epsilon)r$

// where P^v is the path with $min\{w[v,i] |\ i \in [0 \ldots\ldots \lambda]\}$

This algorithm is an extension of the Path Delay Discretization Algorithm (PDA) suggested by Chen *et al*. (2008). Here, it is modified to deal with the intuitionistic fuzzy parameter cost. Two functions $Get\_abcd(\ alpha1, stretch1, cost)$ and $Get\_efgh(\ alpha2, stretch2, cost)$ are used to create TIFN from the randomly generated cost values and TIFNs are ranked using the proposed centroid method of ranking in section 3.2.1 (II) (a). The detailed explanation of the algorithm has been presented in section 2.2.2 (III) (a) (iv) of chapter 2. The analysis of the solution generated from the intuitionistic fuzzy version of Chen's algorithm has been presented in the next section. Since in the intuitionistic fuzzy version, the number of arithmetic operations increase by a constant factor, the worst case complexity of the algorithm remains $O((m + n \log n)L/\varepsilon)$, same as stated by Chen *et al*. (2008).

(b) Experimental Analysis

CIFSPP algorithm was implemented in C language using CodeBlocks for running on an i5 based 3.20 GHz system with 3 GB RAM. Many practical networks like social networks, biological networks, world wide web etc. are *scale-free* networks which follow the power law. Therefore, we generated our test cases using a random graph generator (*gengraph-win*) that follows the power law random graph model. The parameters required to generate a random graph using *gengraph-win* are *n, alpha, min, max, z*. Here, *n* denotes the number of nodes, *min* and *max* represent the minimum degree and maximum degree respectively, *alpha* is a random number ranging from 1-2.5 and denotes the exponent of the power law distribution. Sample graphs with *n* nodes and degrees within the stated range of *min-max* were generated from a heavy-tailed distribution of exponent *alpha* and average *z* using the command "*distrib n* *alpha min max z*". After generating the random graph, crisp cost and delay values within the range of 1 to 100 were assigned to each edge of the graph using the C $rand()$ function. Then these cost values were converted into intuitionistic fuzzy numbers using the $Get\_abcd()$ and $Get\_efgh()$ function of the above stated algorithm. The algorithm was implemented for graphs with different sizes i.e., n = 50, 100, 150, 200 and here we present a plot for a graph with $n = 200$, $alpha = 2.5$, $min = 10$, $max = 20$, $z = 12$, $stretch1 = 2$, $alpha1 = 1$, $stretch2 = 3.5$, $alpha2 = 1$, $source = 15$, $and\ target = 95$. It was observed in all the cases that as the delay constraint was relaxed, the cost of the cheapest path decreased. After a point, the cost became constant (for very large delay constraint value) and was the same as obtained using the classical Dijkstra's algorithm for the stated source-target pair. Fig. 3.3 clearly shows the observation discussed above. As can be seen, that when delay constraint is less than 90, no path is found and beyond 90 as the delay constraint value is increased, the cost starts decreasing and becomes constant beyond delay constraint = 220. We also verified that the results generated by the intuitionistic fuzzy version agrees with those obtained using the crisp case of the algorithm suggested by Chen *et al*. (2008) by setting the width of the intuitionistic fuzzy number representing the cost of the final path equal to zero.

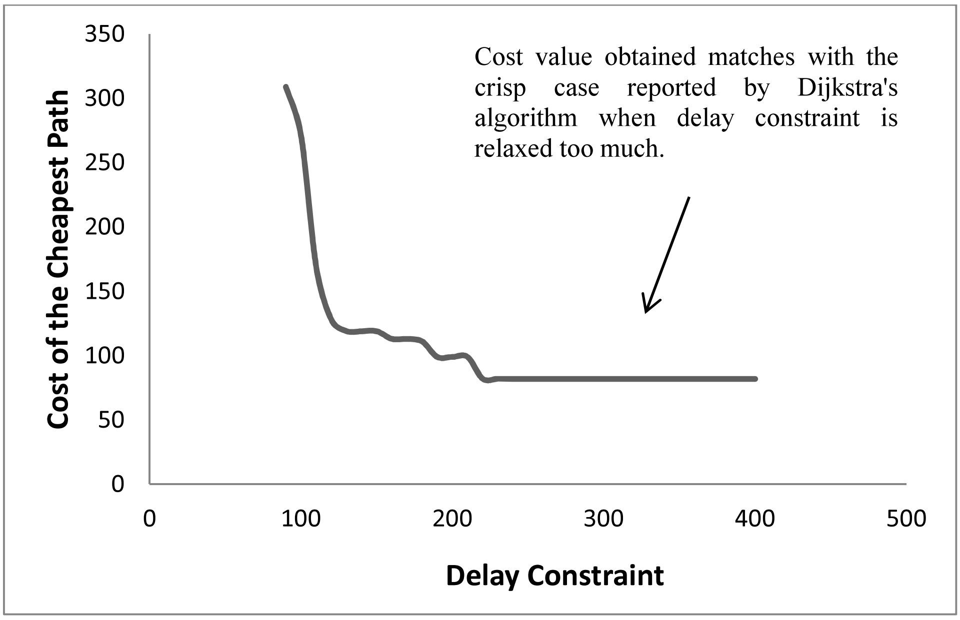


Fig. 3.3: Behaviour shown by the cost of the shortest path on varying the input delay constraint for a graph with 200 nodes generated using *gengraph-win*

3.2.2 Quasi-Gaussian Intuitionistic Fuzzy Number (QGIFN)

Fuzzy numbers can be applied in various real life areas and Gaussian fuzzy numbers (GFN) find a lot of practical applications. Therefore, considering the various fields where GFN can be applied, it is favourable to limit the GFN and define a new variant called the Quasi-Gaussian fuzzy numbers (QGFN) which is a GFN with a restriction that the value of x beyond $\bar{x}$ - $3\sigma_l$ and $\bar{x}$ + $3\sigma_r$ is zero. Here, $\bar{x}$, σ_l and σ_r denotes the modal value, left spread and right spread respectively corresponding to the standard deviation of the Gaussian distribution. The notation used by Hanss (2005) to represent a QGFN is as follows:

$$t = gfn^*(\bar{x}, \sigma_l, \sigma_r)$$

Here t signifies a QGFN and its membership function $\mu_t(x)$ is defined as:

$$\mu_t(x) = \begin{cases} 0 & for\ x \leq \bar{x} - 3\sigma_l \\ \exp\left[-\frac{(x-\bar{x})^2}{2\sigma_l{}^2}\right] & for\ \bar{x} - 3\sigma_l < x < \bar{x} \\ & \forall\ x \in R \\ \exp\left[-\frac{(x-\bar{x})^2}{2\sigma_r{}^2}\right] & for\ \bar{x} \leq x < \bar{x} + 3\sigma_r \\ 0 & for\ x > \bar{x} + 3\sigma_r \end{cases} \quad (3.19)$$

In this chapter, we follow the notation used by Hanss (2005) for QGFN to introduce the Quasi-Gaussian intuitionistic fuzzy number (QGIFN) and use it for modelling the parameter cost involved in CSPP, leading to the intuitionistic version of CSPP called constrained intuitionistic fuzzy shortest path problem (CIFSPP). The definition for the membership and non-membership function of QGIFN is presented below:

$$\mu_t(x) = \begin{cases} 0 & for\ x \leq \bar{x} - 3\sigma_l \\ \exp\left[-\frac{(x-\bar{x})^2}{2\sigma_l{}^2}\right] & for\ \bar{x} - 3\sigma_l < x < \bar{x} \\ & \forall\ x \in R \\ \exp\left[-\frac{(x-\bar{x})^2}{2\sigma_r{}^2}\right] & for\ \bar{x} \leq x < \bar{x} + 3\sigma_r \\ 0 & for\ x > \bar{x} + 3\sigma_r \end{cases} \quad (3.20a)$$

$$\nu_t(x) = \begin{cases} 1 & for\ x \leq \bar{x} - 3\sigma_l' \\ 1 - \exp\left[-\frac{(x-\bar{x})^2}{2\sigma_l{}^2}\right] & for\ \bar{x} - 3\sigma_l' < x < \bar{x} \\ & \forall\ x \in R \\ 1 - \exp\left[-\frac{(x-\bar{x})^2}{2\sigma_r{}^2}\right] & for\ \bar{x} \leq x < \bar{x} + 3\sigma_r' \\ 1 & for\ x > \bar{x} + 3\sigma_r' \end{cases} \quad (3.20b)$$

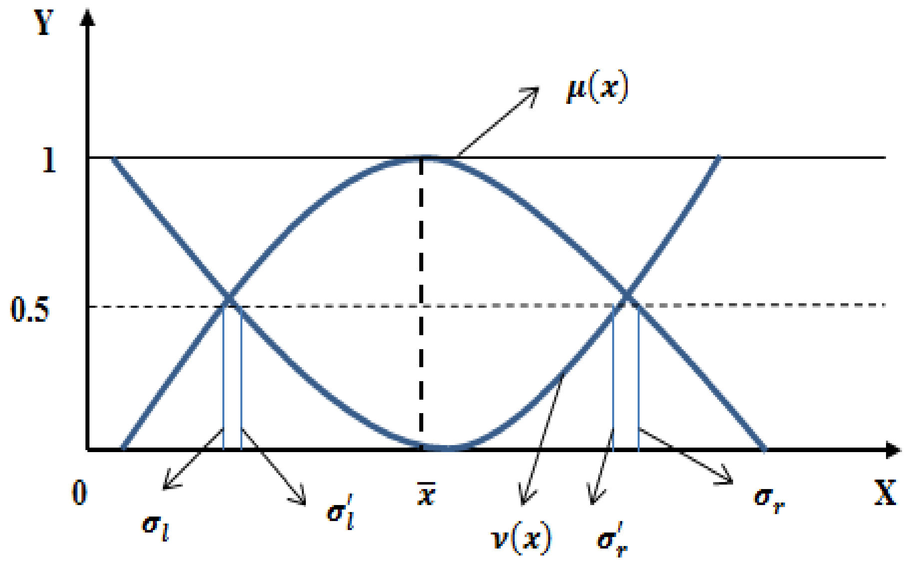


Fig. 3.4: Quasi-Gaussian Intuitionistic Fuzzy Number

(I) Addition of two QGIFNs

Addition of two QGIFNs is performed using decomposed intuitionistic fuzzy numbers which implements interval arithmetic. An intuitionistic fuzzy set Z can be represented as a sequence of $\alpha, \beta - cuts$. An $\alpha, \beta - cut$ of Z can be defined as (Mahapatra, & Roy, 2013):

$$Z_{\alpha,\beta} = \{(x, \mu_Z(x), \nu_Z(x)) : x \in U, \mu_Z(x) \geq \alpha, \nu_Z(x) \leq \beta\} \quad (3.21)$$

where, $\alpha, \beta \in [0,1]$

The infinite number of $\alpha, \beta - cuts$ are reduced to a finite set to make it more appropriate for practical applications by choosing some discrete values $\alpha_j = \mu_j$ for α and $\beta_i = \nu_i$ for β (Hanss, 2005). To create the finite set, the $[0,1]$ interval is divided into k-subintervals and can be denoted using the following notation:

$$Z_s = \left\{\left(X_s^{(0)}, X_s^{(1)}, \dots\dots, X_s^{(k)}\right), \left(X_{s'}^{(k)}, \dots\dots, X_{s'}^{(1)}, X_{s'}^{(0)}\right)\right\} \quad (3.22)$$

Where, $X_s^{(j)} = \left[a_s^{(j)}, b_s^{(j)}\right]$, $X_{s'}^{(i)} = \left[a_{s'}^{(i)}, b_{s'}^{(i)}\right]$ such that $a_s^{(j)} \leq b_s^{(j)}$, $a_{s'}^{(i)} \leq b_{s'}^{(i)}$, $j + i \leq 1$.

If we have two decomposed intuitionistic fuzzy numbers $X_1^{(j,i)} = \left(\left[a_1^{(j)}, b_1^{(j)}\right], \left[a_{1'}^{(i)}, b_{1'}^{(i)}\right]\right)$ and $X_2^{(j,i)} = \left(\left[a_2^{(j)}, b_2^{(j)}\right], \left[a_{2'}^{(i)}, b_{2'}^{(i)}\right]\right)$, then their addition can be performed using the following formula:

$$X = X_1^{(j,i)} + X_2^{(j,i)} = \left(\left[a_1^{(j)}, b_1^{(j)}\right], \left[a_{1'}^{(i)}, b_{1'}^{(i)}\right]\right) + \left(\left[a_2^{(j)}, b_2^{(j)}\right], \left[a_{2'}^{(i)}, b_{2'}^{(i)}\right]\right)$$

$$= \underbrace{\left(\left[a_1^{(j)} + a_2^{(j)}, b_1^{(j)} + b_2^{(j)}\right], \left[a_{1'}^{(i)} + a_{2'}^{(i)}, b_{1'}^{(i)} + b_{2'}^{(i)}\right]\right)}_{\left(\left[a^{(j)}, b^{(j)}\right], \left[a'^{(i)}, b'^{(i)}\right]\right)} \quad (3.23)$$

(II) Ranking of QGIFNs

The general method for calculating the centroid of a curved area utilizes the following formulas:

Centroid $(C) = (x_c, y_c)$

Here, $x_c = \frac{total\ moments\ in\ X-direction}{total\ area}$ and $y_c = \frac{total\ moments\ in\ Y-direction}{total\ area}$.

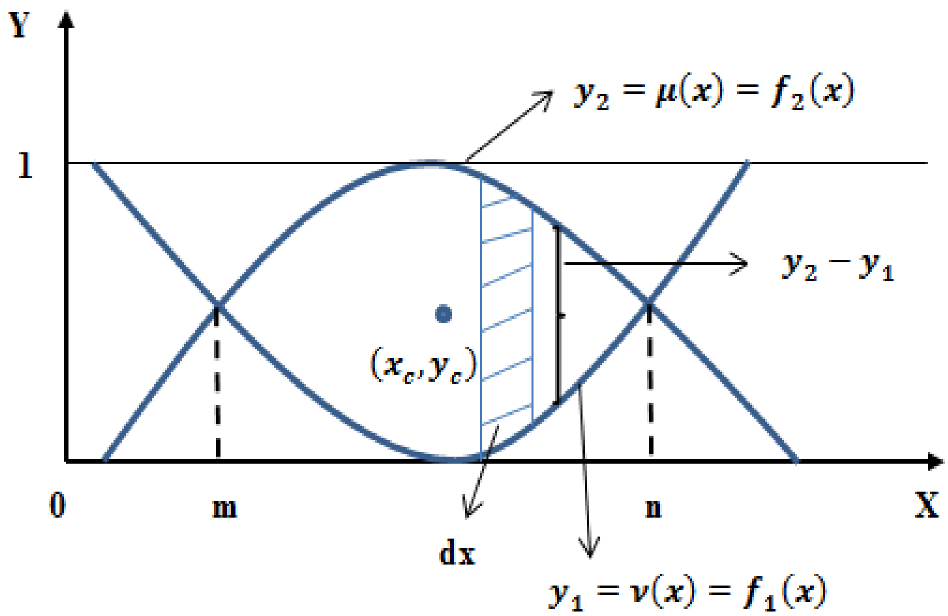


Fig. 3.5: Centroid method of ranking for QGIFN

We extend the general formula for computing the centroid to be applied on QGIFN. As shown in Fig. 3.5, the total moments in the X and Y direction over the total area can be stated as (Marghitu & Dupac, 2012):

$$x_c = \frac{total\ moments\ in\ X\!-\!direction}{total\ area} = \frac{1}{A}\int_m^n x(y_2 - y_1)\,dx \tag{3.24}$$

$$y_c = \frac{total\ moments\ in\ Y\!-\!direction}{total\ area} = \frac{1}{A}\int_m^n \frac{[y_2]^2 - [y_1]^2}{2}\,dx \tag{3.25}$$

Integrating a bell shaped curve or a Gaussian curve is not possible by using simple mathematical techniques. Therefore, few advanced techniques like series expansion and *integration as summation* is used to integrate these bell shaped and Gaussian curves. Here, we prefer the *integration as summation* method to integrate the Quasi-Gaussian curve.

We assume $dx = 0.000001(STEP\_SIZE)$ where $STEP\_SIZE$ is a minute value and to integrate the curve, the following formula is used:

$$\int_m^n f(x)\,dx = \lim_{n\to\infty} \sum_{i=1}^{n} f(x_i^*)\,\Delta x_i \tag{3.26}$$

Also, we consider the critical cases where the value of spreads (either left or right) is zero. For the computation ease and to remove the complexities that may arise due to the zero value, the spread (either left or right) that is zero is replaced by a very small ϵ value.

After the centroid (C) is obtained, the distance (D) is computed using the following formula:

$$D = \sqrt{(x_c)^2 + (y_c)^2} \tag{3.27}$$

Then the paths are ranked (R) according to their distance (D) values, that is a path having lesser (D) value gets a higher ranking and vice versa. If we have two QGIFNs A and B with the same distance (D) value then they are ranked according to their spreads i.e., the number with a smaller spread gets a higher ranking (R).

(III) Application

(a) CIFSPP Algorithm (using QGIFN)

$\boldsymbol{Get\_abcde}(\ \boldsymbol{cost}, \boldsymbol{\sigma_l}, \boldsymbol{\sigma_r}, \boldsymbol{\sigma'_l}, \boldsymbol{\sigma'_r})$

1. $b = cost$
2. $a = b - 3 * \sigma_l$
3. $c = b + 3 * \sigma_r$
4. $d = b - 3 * \sigma'_l$
5. $e = b + 3 * \sigma'_r$

$\boldsymbol{Initialize}\ (\boldsymbol{V}, \boldsymbol{s}, \boldsymbol{\lambda})$

```
1.for each vertex v ϵ V, each i ϵ [0, … … , λ]
2.    Get_abcde(cost, σ_l, σ_r, σ'_l, σ'_r)
3.    w[v, i] ≔ ∞, π[v, i] ≔ NIL, z[v, i] ≔ ∞
4.    w[s, 0] ≔ 0, z[s, i] ≔ 0
5.end for
```

$\boldsymbol{Relax\_FPDA}(\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{i}, \boldsymbol{\lambda})$

```
1.  i' ≔ floor((z[u,i] + d(u,v))/r λ)
2.  Get_abcde( cost, σ_l, σ_r, σ'_l, σ'_r)
3.  if i' ≤ λ and w[v, i'] > w[u, i] + c(u, v)
// Compare using Equation 3.24,3.25,3.26,3.27 of section 3.2.2 (II).
4.          w[v, i'] ≔ w[u, i] + c(u, v)
5.          π[v, i'] ≔ u
6.          z[v, i'] ≔ min{z[v, i'],  z[u, i] + d(u, v)}
7.  end if
```

$\boldsymbol{FPDA\_Dijkstra}(\boldsymbol{G}, \boldsymbol{s}, \boldsymbol{\lambda})$

```
1.Initialize (V, s, λ)
2.  for i = 0 to λ
3.        Q ≔ V
4.        while Q ≠ φ
```

$u \coloneqq Extract\_Min(Q)$

$\boldsymbol{if}\ w[u, i] = \infty$

$\boldsymbol{break}\ out\ of\ the\ \boldsymbol{while}\ loop$

$\boldsymbol{end\ if}$

$Q \coloneqq Q - \{\, u \,\}$

$\boldsymbol{for}\ every\ adjacent\ node\ \boldsymbol{v}\ of\ \boldsymbol{u}$

$\boldsymbol{Relax\_FPDA}(\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{i}, \boldsymbol{\lambda})$

$\boldsymbol{end\ for}$

$\boldsymbol{end\ while}$

$\boldsymbol{end\ for}$

$\boldsymbol{FPDA}(\boldsymbol{G}, \boldsymbol{s})$

$\lambda \coloneqq \lambda_0$

$\boldsymbol{do}$

$\lambda \coloneqq 2\lambda$

$\boldsymbol{FPDA\_Dijkstra}(\boldsymbol{G}, \boldsymbol{s}, \boldsymbol{\lambda})$

$\boldsymbol{while}\ \exists v \in d(P^v) > (1 + \epsilon) r$

// where P^v is the path with $min\{w[v, i] \mid i \in [0 \ldots\ldots \lambda]\}$

The above stated algorithm is the extended version of the path delay discretization algorithm (PDA) suggested by Chen *et al*. (2008), which is capable of tackling the intuitionistic fuzzy environment. Amongst the two parameters cost and delay of constrained intuitionistic fuzzy shortest path problem (CIFSPP), cost is represented as a Quasi-Gaussian intuitionistic fuzzy number (QGIFN). In this algorithm, the function $Get\_abcde\ (cost, sigma1, sigma2, sigma3, sigma4)$ generates the QGIFN using the randomly generated cost and sigma values. A detailed explanation of the algorithm has been stated in section 2.2.2. (III) (a) (iv) of chapter 2. The analysis of the intuitionistic version of PDA has been presented in the next section. The worst case complexity of the above algorithm remains the same as stated by Chen *et al*. (2008) i.e., $O((m + n \log n)L/\varepsilon)$ where L signifies the longest path in the network.

(b) Experimental Analysis

The CIFSPP algorithm has been implemented in C language and compiled using CodeBlocks for running on an i7 based system running at 3.40 GHz with 4GB RAM. The algorithm was implemented for large test graphs generated using the power law random graph generator *gengraph-win.* Initially, using the command "*distrib n alpha min max z*" several unweighted power law sample graphs with a random structure were generated with different network sizes i.e., different number of nodes like 50, 100, 150, 200, 225, 250 etc. Then random weights were assigned to the edges of the graph from the uniformly distributed random numbers ranging from 1 to 100 created using C *rand()* function. After that QGIFNs were generated for the cost values using the function $Get\_abcde()$ of the CIFSPP algorithm. To show the behaviour of the output cost with the varying input delay constraint (Fig. 3.6), we include the plot of a graph with $n = 100,\ alpha = 2.5,\ min = 7,\ max = 15,\ z = 10,\ \text{source} = 23 \text{ and target} = 75$. By repeating the experiments sufficient number of times, it was observed that on relaxing the delay constraint, the cost of the cheapest path decreases. The reason for this behaviour is that if the delay constraint is strict, the algorithm does not find enough time to obtain the best possible path however, among the feasible paths, the cheapest one is selected. On relaxing the delay constraint to some extent, it was observed that the cost of the best path decreases because the algorithm found more time to explore the solution space. Therefore, a cheaper path could be computed. Although beyond a point (for a very large delay constraint value), even after relaxing the delay constraint, the cost does not decrease, becomes constant and is the same as obtained by the Classical Dijkstra's algorithm i.e., no cheaper path than this is available for the stated source-target pair.

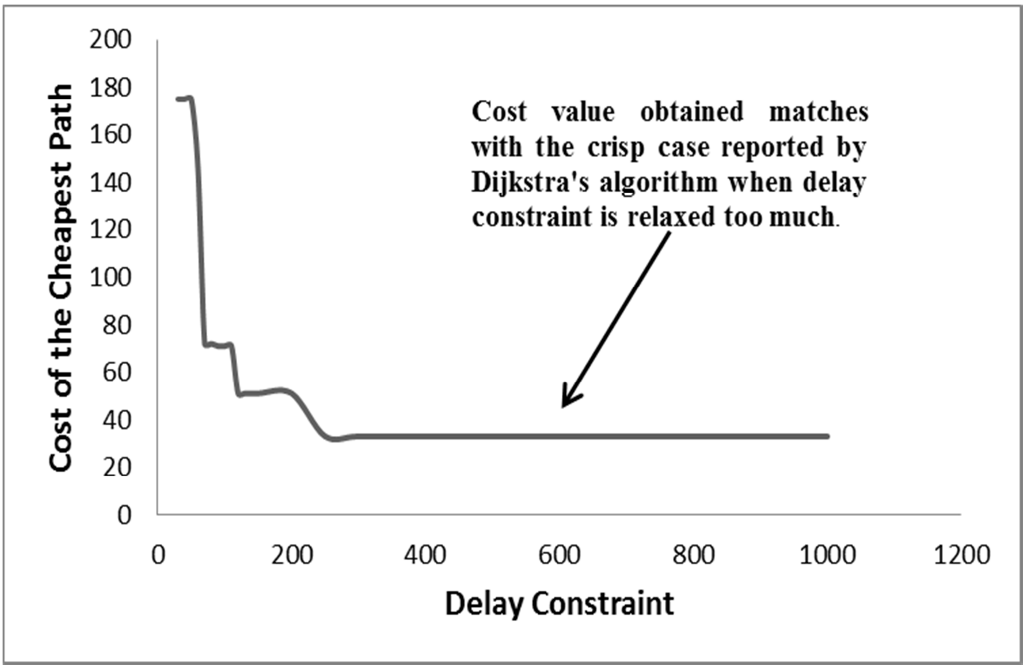


Fig. 3.6: Trend observed in the cost of the shortest path on varying the input delay constraint for a graph with 100 nodes generated using gengraph-win

3.3 Max – Min Formulation for Intuitionistic Fuzzy Orienteering Problem

Here, we introduce the intuitionistic fuzzy orienteering problem (IFOP) where the parameters (score and time) are represented using TIFN. In IFOP, the strict requirements of the crisp formulation which include the maximization or minimization of the objective function, satisfying each and every constraint and giving equal importance to all the constraints are relaxed to some extent by using fuzzy logic with the aim to provide a more accurate and practical modeling of the real world. In the fuzzy formulation we consider the willingness of the decision maker, his aspiration levels and the degree up to which a solution is acceptable or its degree of satisfaction. This modeling is made more apt by using intuitionistic fuzzy numbers as the degree of non-belongingness is also considered along with the degree of belongingness and this is the best way to tackle the vagueness (Zimmermann, 2010).

3.3.1 IFOP Algorithm

Following are the steps to determine the most appropriate path for a given graph $G(V,E)$ with N nodes. The steps are explained in the next section with the help of an illustrative example:

Step 1: Compute all the paths (P_m) that connect the source node (v_1) and the destination node (v_N) and fulfil the condition stated by the Eqs. 2.45, 2.46, 2.47, 2.49, 2.50 and 2.51 of chapter 2.

Step 2: The following values are calculated for each of the possible paths computed in Step 1**:**

(a) The total collected reward or score and the total time taken (using Eq. 3.2 of section 3.2.1 (I)).

(b) The expected value for the total time taken and the total collected score using the following formula:

For a given TIFN $A = \langle (a,b,c,d),(e,f,g,h) \rangle$

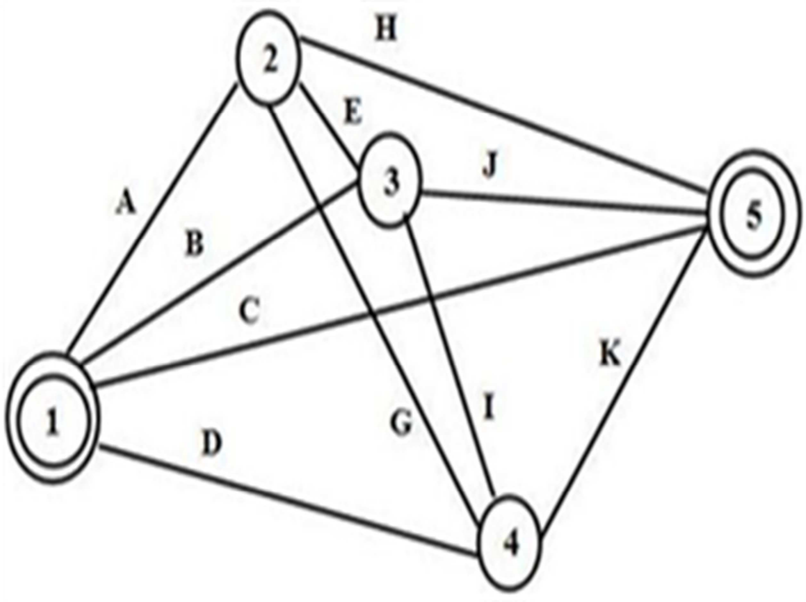
$$EV(A) = \frac{1}{8}(a+b+c+d+e+f+g+h) \tag{3.28}$$

(c) The membership value for the total time taken and the total collected score represented by the fuzzy set $F_1$and F_2respectively (using Eq. 2.48 and Fig. 2.22 and Eq. 2.44 and Fig. 2.21 of chapter 2).

Step 3: Compute the set of feasible paths depicted by the fuzzy decision set Z (using definition in section 2.3.1 (b) and Eq. 2.42 of chapter 2).

Step 4: The final solution representing the most desirable path is denoted by the fuzzy decision set Z^* (obtained using definition in section 2.3.1 (b) and Eq. 2.43 of chapter 2). If the set Z^* contains more than one path then to conclude with the path that maximizes the total collected score, the paths in Z^* are ranked according to their total collected score (using Eqs. 3.13, 3.14, 3.15).

3.3.2 Illustrative Example



Node	Label	Intuitionistic Fuzzy Score Values $\langle(\mu(x)),(\nu(x))\rangle$
v_1	1	$\langle(1,2,8,9),(0,2,8,10)\rangle$
v_2	2	$\langle(8,9,11,12),(6,8,12,14)\rangle$
v_3	3	$\langle(3,5,9,11),(1,4,10,13)\rangle$
v_4	4	$\langle(17,20,24,27),(14,18,26,30)\rangle$
v_5	5	$\langle(1,2,4,5),(0,1,5,6)\rangle$
	$S_{min}=25, P=13$	

Edge	Label	Intuitionistic Fuzzy Time Values $\langle(\mu(x)),(\nu(x))\rangle$
e_{12}	A	$\langle(1,4,6,9),(0,2,8,10)\rangle$
e_{13}	B	$\langle(6,9,13,16),(5,8,14,17)\rangle$
e_{14}	D	$\langle(14,15,21,22),(12,14,22,24)\rangle$
e_{15}	C	$\langle(4,6,8,10),(2,4,10,12)\rangle$
e_{23}	E	$\langle(2,3,3,4),(0,3,3,6)\rangle$
e_{24}	G	$\langle(5,8,8,11),(4,8,8,12)\rangle$
e_{25}	H	$\langle(14,18,22,26),(12,16,24,28)\rangle$
e_{34}	I	$\langle(5,8,12,15),(3,6,14,17)\rangle$
e_{35}	J	$\langle(0,2,10,12),(0,1,11,12)\rangle$
e_{45}	K	$\langle(1,2,2,3),(0,2,2,4)\rangle$
	$T_{max}=20, L=15$	

Fig. 3.7: The input graph with $N=5, v_1=1, v_N=5$ and the time and score values associated with each edge and vertex respectively

Step1: For the given graph G, following are the possible paths:

$\boldsymbol{P_1}: 1-5$;

$\boldsymbol{P_2}: 1-2-5$;

$\boldsymbol{P_3}: 1-3-5$;

$\boldsymbol{P_4}: 1-4-5$;

$\boldsymbol{P_5}: 1-2-3-5$;

$\boldsymbol{P_6}: 1-3-2-5$;

$\boldsymbol{P_7}: 1-2-4-5$;

$\boldsymbol{P_8}: 1-4-2-5$;

$\boldsymbol{P_9}: 1-3-4-5$;

$\boldsymbol{P_{10}}: 1-4-3-5$;

$\boldsymbol{P_{11}}: 1-2-3-4-5$;

$\boldsymbol{P_{12}}: 1-2-4-3-5$;

$\boldsymbol{P_{13}}: 1-3-4-2-5$;

$\boldsymbol{P_{14}}: 1-4-3-2-5$;

$\boldsymbol{P_{15}}: 1-4-2-3-5$;

$\boldsymbol{P_{16}}: 1-3-2-4-5$

Step 2: The actual values for the stated example obtained as a result of step 2 (a), (b), (c) of the IFOP algorithm are shown in the Table 3.1 (a), (b), (c).

Table 3.1(a): The values of total time taken and total collected score obtained for each possible path

Path	Total Time Taken	Total Collected Score
P_1	$\langle(4,6,8,10),(2,4,10,12)\rangle$	$\langle(1,2,8,9),(0,2,8,10)\rangle$
P_2	$\langle(15,22,28,35),(12,18,32,38)\rangle$	$\langle(9,11,19,21),(6,10,20,24)\rangle$
P_3	$\langle(6,11,23,28),(5,9,25,29)\rangle$	$\langle(4,7,17,20),(1,6,18,23)\rangle$
P_4	$\langle(15,17,23,25),(12,16,24,28)\rangle$	$\langle(18,22,32,36),(14,20,34,40)\rangle$
P_5	$\langle(3,9,19,25),(0,6,22,28)\rangle$	$\langle(12,16,28,32),(7,14,30,37)\rangle$
P_6	$\langle(22,30,38,46),(17,27,41,51)\rangle$	$\langle(12,16,28,32),(7,14,30,37)\rangle$
P_7	$\langle(7,14,16,23),(4,12,18,26)\rangle$	$\langle(26,31,43,48),(20,28,46,54)\rangle$
P_8	$\langle(33,41,51,59),(28,38,54,64)\rangle$	$\langle(26,31,43,48),(20,28,46,54)\rangle$
P_9	$\langle(12,19,27,34),(8,16,30,38)\rangle$	$\langle(21,27,41,47),(15,24,44,53)\rangle$
P_{10}	$\langle(19,25,43,49),(15,21,47,53)\rangle$	$\langle(21,27,41,47),(15,24,44,53)\rangle$
P_{11}	$\langle(9,17,23,31),(3,13,27,37)\rangle$	$\langle(29,36,52,59),(21,32,56,67)\rangle$
P_{12}	$\langle(11,22,36,47),(7,17,41,51)\rangle$	$\langle(29,36,52,59),(21,32,56,67)\rangle$
P_{13}	$\langle(30,43,55,68),(24,38,60,74)\rangle$	$\langle(29,36,52,59),(21,32,56,67)\rangle$
P_{14}	$\langle(35,44,58,67),(27,39,63,75)\rangle$	$\langle(29,36,52,59),(21,32,56,67)\rangle$
P_{15}	$\langle(21,28,42,49),(16,26,44,54)\rangle$	$\langle(29,36,52,59),(21,32,56,67)\rangle$
P_{16}	$\langle(14,22,26,34),(9,21,27,39)\rangle$	$\langle(29,36,52,59),(21,32,56,67)\rangle$

Table 3.1(b): The expected value for the total time taken and the total collected score of each possible path

Path	EV (Total Time Taken)	EV (Total Collected Score)
P_1	7	5
P_2	25	15
P_3	17	12
P_4	20	27
P_5	14	22
P_6	34	22
P_7	15	37
P_8	46	37
P_9	23	34
P_{10}	34	34
P_{11}	20	44
P_{12}	29	44
P_{13}	49	44
P_{14}	51	44
P_{15}	35	44
P_{16}	24	44

Table 3.1(c): The membership value for the total time taken and the total collected score for each possible path

Path	$\mu_T(P_m)$	$\mu_S(P_m)$
P_1	1	0
P_2	0.67	0.23
P_3	1	0
P_4	1	1
P_5	1	0.77
P_6	0.06	0.77
P_7	1	1
P_8	0	1
P_9	0.8	1
P_{10}	0.06	1
P_{11}	1	1
P_{12}	0.4	1
P_{13}	0	1
P_{14}	0	1
P_{15}	0	1
P_{16}	0.73	1

The fuzzy set F_1 and F_2 (as shown below) denotes the membership value for the total time taken and the total collected score respectively:

$$F_1 = \{P_1/1\,, P_2/0.67, P_3/1, P_4/1\,, P_5/1, P_6/0.06, P_7/1, P_8/0, P_9/0.8, P_{10}/0.06, P_{11}/1, P_{12}/0.4, P_{13}/0\,, P_{14}/0, P_{15}/0, P_{16}/0.73\}$$

$$F_2 = \{P_1/0\,, P_2/0.23, P_3/0, P_4/1\,, P_5/0.77, P_6/0.77, P_7/1, P_8/1, P_9/1, P_{10}/1, P_{11}/1, P_{12}/1, P_{13}/1\,, P_{14}/1, P_{15}/1, P_{16}/1\}$$

Step 3: For the considered input, following is the fuzzy decision Z:

$$Z = \{P_1/0\,, P_2/0.23, P_3/0, P_4/1\,, P_5/0.77, P_6/0.06, P_7/1, P_8/0, P_9/0.8,$$
$$P_{10}/0.06, P_{11}/1, P_{12}/0.4, P_{13}/0\,, P_{14}/0, P_{15}/0, P_{16}/0.73\}$$

Step 4: Following are the paths in Z^* and their corresponding ranks obtained for the given network:

$$Z^* = \{P_4/1\,, P_7/1\,, P_{11}/1\}$$

Table 3.2: Ranks of the desirable paths

Path	Score	Rank
P_4	$\langle(18,22,32,36),(14,20,34,40)\rangle$	3
P_7	$\langle(26,31,43,48),(20,28,46,54)\rangle$	2
$\boldsymbol{P_{11}}$	$\boldsymbol{\langle(29,36,52,59),(21,32,56,67)\rangle}$	**1**

The most desirable path is P_{11} as it has the highest rank. To check the correctness of our result, we set the spreads of the TIFN (for both score and time) to zero in order to convert each input to its crisp equivalent and then perform exhaustive search. This gives the same solution as our algorithm.

3.3.3 Work-depth Analysis of IFOP

We present the work-depth analysis of IFOP here, to achieve a better performance and solve it more efficiently for large instances. There are several ways for organizing parallel computers. Therefore, it becomes difficult to determine one multiprocessor model that is apt for all machines. The method to deal with this situation is to focus on algorithms than on machines. As stated in (Blelloch, 1996), work-depth model is a technique of modelling the parallelism of an algorithm. For any algorithm, the following terms can be calculated (Blelloch, 1996):

Work (W): Total number of operations performed.

Depth (D): Longest chain of dependencies among its operations.

Parallelism ($\mathcal{P}$): The ratio $\frac{W}{D}$

Algorithms with efficient work-depth models can be converted into efficient multiprocessor models and then to actual parallel computers. Work-depth models can be represented in three possible ways:

- Circuit Model.
- Vector Machine Model.
- Language-based Model.

For the parallel formulation of IFOP we use the circuit model which is the most abstract one when compared to the other two models. A circuit has two important components: nodes and directed arcs. The directed arcs and the nodes denote the flow of values and the operations to be performed respectively. Fan-in and fan-out are two terms associated with each node signifying the number of incoming and outgoing arcs respectively. The input to the circuit is provided through input arcs which do not originate from any node. Similarly, the output arcs carry the result out of the circuit and do not have any destination node. The number of nodes denotes the work of the circuit, also called the size of the circuit. A circuit should not contain directed cycles and the count of the nodes on the longest directed path connecting the input and output arc specifies the depth of the circuit. If the parallelism computed for the work-depth model is at least as large as the number of processors, then it is said to be work-preserving and can be translated into an efficient multiprocessor model like the PRAM model (Blelloch, 1996).

For the circuit of IFOP we assume two things:

$(a)\ number\ of\ processors =$
$Number\ of\ distinct\ paths\ between\ source\ (v_1)\ and$ destination (v_N).

(b) *number of distinct paths* $(p) =$

$$\sum_{i=1}^{N-1} \frac{(N-2)!}{[N-(i+1)]!} \quad \textit{where } N \textit{ is the number of nodes in the given graph } G.$$

The first step of the IFOP of determining all the possible distinct paths is performed sequentially as shown below:

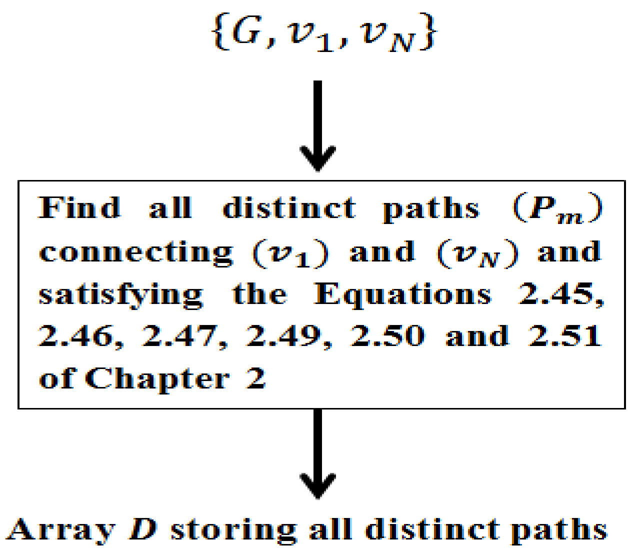


Fig. 3.8: The sequential module executing step 1 of IFOP that computes all the distinct paths in the given graph *G*

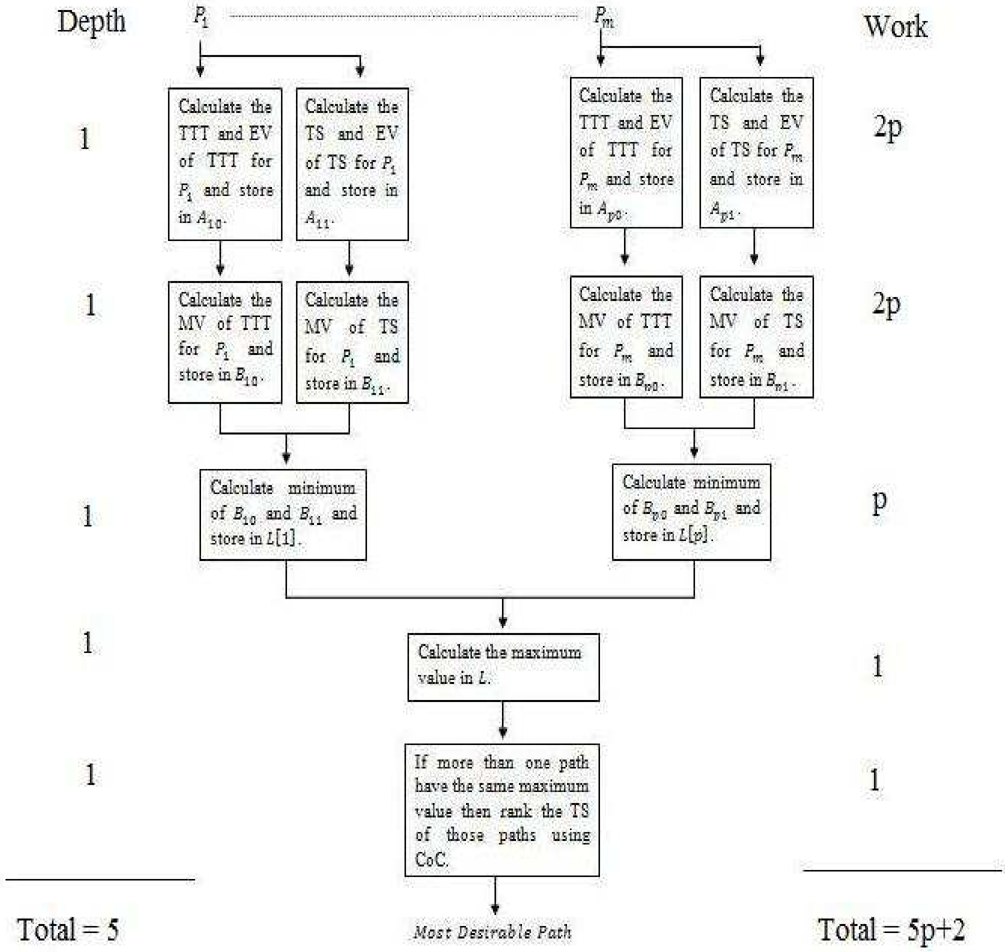
Fig 3.9 presents the circuit of IFOP. The parallelism for IFOP is as follows:

Work (W) = $5p + 2$.

Depth (D) = 5.

Parallelism ($\mathcal{P}$) $= \frac{W}{D} = \frac{5p+2}{5} = \left(p + \frac{2}{5}\right)$.

Therefore, IFOP is work-preserving because the parallelism ($\mathcal{P}$) is $\left(p + \frac{2}{5}\right)$ which is at least as large as the number of processors viz. p. Thus, it is seen that the intuitionistic fuzzy formulation of the orienteering problem is efficiently parallelizable.



TTT = Total Travel Time.
TS = Total Score.
EV = Expected Value.
MV = Membership Value.
A = A two dimensional array of size p x 2.
B = A two dimensional array of size p x 2.
L = A one dimensional array of size p.

Fig. 3.9: The parallel version of IFOP along with its work-depth analysis stating the work and depth value of each step

3.4 Intuitionistic Fuzzy Metric Space using Intuitionistic Fuzzy Point

In this chapter, we use the concept of IFP and intuitionistic fuzzy scalars for proposing a new definition of the intuitionistic fuzzy metric space. The proposed definition of distance metric is then applied to the orienteering problem resulting in the formulation of the intuitionistic fuzzy orienteering problem (IFOP) where for a given graph $G(V, E)$, the task is to compute a Hamiltonian path P that connects the stated source (v_1) and target (v_N) along with a subset of vertices (V') of the vertex set V and also satisfies the upper limit on the distance covered (D_{max}). The intuitionistic fuzzy version of the problem has been considered for the first time and we deal with both, the uncertainty in the parameter score using trapezoidal intuitionistic fuzzy number and the uncertainty in the position / location of the point of interest using intuitionistic fuzzy points. Further, an algorithm for solving IFOP has been suggested in section 3.3.1 of this chapter. An intuitionistic fuzzy point $c(\lambda, \alpha)$ can be defined in the following form (Coker & Demirci, 1995):

$$c(\lambda, \alpha) = \{\langle x, c_\lambda, 1 - c_{1-\alpha}\rangle | x \in U\}$$

(Here, $c \in U$ is the support of $c(\lambda, \alpha)$ and λ and α represents the degree of membership and degree of non-membership respectively of $c(\lambda, \alpha)$).

U is a nonempty set and $c \in U$. $\lambda \in (\,0, 1\,]$ and $\alpha \in [\,0\,, 1)$ are two real numbers such that $\lambda + \alpha \leq 1$.

In this chapter, the notation used to signify an intuitionistic fuzzy point is (x, λ, i). $P_{IF}(U)$ denotes the set of all intuitionistic fuzzy points defined on U. When, $U = R$, intuitionistic fuzzy points are called intuitionistic fuzzy scalars and $S_{IF}(R)$ denotes the set of all intuitionistic fuzzy scalars. An intuitionistic fuzzy point is said to belong to an intuitionistic fuzzy set K if

$$K = \{(x, \lambda, i) | \mu_K(x) \geq \lambda, \nu_K(x) \leq i\,\}$$

3.4.1 Intuitionistic Fuzzy Metric Space

In this section a few necessary definitions are presented:

Definition 1: Let (a, λ, i) and (b, γ, j) be two intuitionistic fuzzy scalars then we say that:

(1) $(a, \lambda, i) \succcurlyeq (b, \gamma, j)$ if $a > b$ or $(a, \lambda, i) = (b, \gamma, j)$.

(2) (a, λ, i) is said to be no less than (b, γ, j) if $a \geq b$ denoted by $(a, \lambda, i) \succ (b, \gamma, j)$ or $(b, \gamma, j) \prec (a, \lambda, i)$.

(3) (a, λ, i) is said to be non-negative if $a \geq 0$. The set of all the non-negative intuitionistic fuzzy scalars is denoted by $S_{IF}^{+}(R)$.

Here, both the operators $\succ$ and $\succcurlyeq$ denote partial ordering.

Definition 2: Let U be a nonempty set and $d_{IF}: P_{IF}(U)\ X\ \ P_{IF}(U) \to S_{IF}^{+}(R)$ be a mapping. For any $\{(x, \lambda, i), (y, \gamma, j), (z, \rho, l)\} \subset P_{IF}(U)$, if d_{IF} satisfies the following three conditions:

(1) ***Non Negative:*** $d_{IF}\big((x, \lambda, i), (y, \gamma, j)\big) \geq 0$ and $d_{IF}\big((x, \lambda, i), (y, \gamma, j)\big) = 0$ iff $x = y,\ \lambda = \gamma = 1, i = j = 0$ (in $d_{IF}\big((x, \lambda, i), (y, \gamma, j)\big) = 0$, 0 denotes the intuitionistic fuzzy scalar with membership degree 1 and non-membership degree 0).

(2) ***Symmetric:*** $d_{IF}\big((x, \lambda, i), (y, \gamma, j)\big) = d_{IF}\big((y, \gamma, j), (x, \lambda, i)\big)$.

(3) ***Triangle Inequality:***

$$d_{IF}\big((x, \lambda, i), (z, \rho, l)\big) \prec d_{IF}\big((x, \lambda, i), (y, \gamma, j)\big) + d_{IF}\big((y, \gamma, j), (z, \rho, l)\big) .$$

Here, the summation is defined as:

$$(x, \lambda, i) + (y, \gamma, j) = (x + y, min\{\lambda, \gamma\}, max\{i, j\}) .$$

Then, $(P_{IF}(U), d_{IF})$ is called an intuitionistic fuzzy metric space. Here, $(x, \lambda, i), (y, \gamma, j), (z, \rho, l)$ are intuitionistic fuzzy points, d_{IF} is the intuitionistic fuzzy metric defined in $P_{IF}(U)$ and $d_{IF}\big((x, \lambda, i), (y, \gamma, j)\big)$ is the intuitionistic fuzzy distance between two intuitionistic fuzzy points.

Proposition 1: Let (U, d) be an ordinary metric space. If (x, λ, i) and (y, γ, j) are two intuitionistic fuzzy points in $P_{IF}(U)$, then we define the distance between them as

$$d_{IF}\big((x, \lambda, i), (y, \gamma, j)\big) = (d(x, y), min\{\lambda, \gamma\}, max\{i, j\})$$

Here, $d(x, y)$ denotes the distance between x and y defined in (X, d). Therefore, $(P_{IF}(U),\ d_{IF})$ is an intuitionistic fuzzy metric space if it satisfies the three necessary conditions stated in Definition 2 of section 3.4.1.

Proof: Here we show that d_{IF} obeys the three conditions given in Definition 2 of section 3.4.1:

(a) ***Non-Negative*****:** Let (x, λ, i) and (y, γ, j) be two IFPs in $P_{IF}(U)$. If $d(x, y)$ is the distance between x and y then we can say $d(x, y) \geq 0$.
From Definition 1 of section 3.4.1, it follows that $d_{IF}\big((x, \lambda, i), (y, \gamma, j)\big) = (d(x, y), min\{\lambda, \gamma\}, max\{i, j\})$ is a non-negative intuitionistic fuzzy scalar and $d_{IF}\big((x, \lambda, i), (y, \gamma, j)\big) = 0$ iff $d(x, y) = 0,\ min\{\lambda, \gamma\} = 1,\ max\{i, j\} = 0$ i.e., $x = y,\ \lambda = \gamma = 1,\ i = j = 0$.

(b) ***Symmetric*****:** Let (x, λ, i) and (y, γ, j) be two IFPs in $P_{IF}(U)$, then we have

$$\begin{aligned} d_{IF}\big((x, \lambda, i), (y, \gamma, j)\big) &= (d(x, y), min\{\lambda, \gamma\}, max\{i, j\}) \\ &= (d(y, x), min\{\gamma, \lambda\}, max\{j, i\}) \\ &= d_{IF}\big((y, \gamma, j), (x, \lambda, i)\big). \end{aligned}$$

(c) ***Triangle Inequality*****:** Let $(x, \lambda, i), (y, \gamma, j)\ and\ (z, \rho, l)$ be three IFPs, then

$$\begin{aligned} d_{IF}\big((x, \lambda, i), (z, \rho, l)\big) &= (d(x, z), min\{\lambda, \rho\}, max\{i, l\}) \\ &\prec (d(x, y) + d(y, z), min\{\lambda, \rho, \gamma\}, max\{i, l, j\}) \\ &= (d(x, y), min\{\lambda, \gamma\}, max\{i, j\}) + \ (d(y, z), min\{\gamma, \rho\}, max\{j, l\}) \\ &= d_{IF}\big((x, \lambda, i), (y, \gamma, j)\big) + d_{IF}\big((y, \gamma, j), (z, \rho, l)\big) \end{aligned}$$

Proposition 2: Let R^n be the $n-$dimensional Euclidean space and T an intuitionistic fuzzy linear space defined in R^n. Suppose (x, λ, i) and (y, γ, j) be two arbitrary intuitionistic fuzzy points $belonging\ to\ T$, then the distance between them can be defined as:

$$d_{IFE}\big((x, \lambda, i), (y, \gamma, j)\big) = (d_E(x, y), min\{\lambda, \gamma\}, max\{i, j\})$$

Here, d_E denotes the Euclidean distance. Therefore, $(T,\ d_{IFE})$ is also an intuitionistic fuzzy metric space, where T can be considered as the set of intuitionistic fuzzy points that $belong\ to$ the intuitionistic fuzzy set T.

Proof: In the ordinary sense, R^n is a metric space and T can be viewed as a subset of $P_{IF}(R^n)$. Therefore, d_{IFE} is an intuitionistic fuzzy metric. (from the proof of Proposition 1).

$S_{IF}^{+}(R)$ is not a complete ordered set. Therefore, in Definition 2 of section 3.4.1 for triangle inequality $\leq$ is replaced by $\prec$. $\prec$ is much weaker than $\leq$, so the natural query that arises is as follows:

Is there some kind of intuitionistic fuzzy metric space that can satisfy the triangle inequality with some partial order which is stronger than $\prec$ like $\preccurlyeq$.

The answer to the query is YES and such kind of metric spaces are called strong intuitionistic fuzzy metric spaces.

Definition 3: Let U be a nonempty set and d_{IF}: $P_{IF}(U)\ X\ \ P_{IF}(U) \rightarrow S_{IF}^{+}(R)$ be a mapping. $(P_{IF}(U),\ d_{IF})$ can be called a strong intuitionistic fuzzy metric space if it fulfils the first two conditions of Definition 2 in section 3.4.1 and for any $(x, \lambda, i), (y, \gamma, j)\ and\ (z, \rho, l)$ in $P_{IF}(U)$ we have:

(3′) $d_{IF}\big((x, \lambda, i), (z, \rho, l)\big) \preccurlyeq\ d_{IF}\big((x, \lambda, i), (y, \gamma, j)\big) + d_{IF}\big((y, \gamma, j), (z, \rho, l)\big)$

Proposition 3: Suppose T is an intuitionistic fuzzy linear space defined in R^n, then the distance between any two intuitionistic fuzzy points on T can be stated as:

$$d_{IFE}\big((x,\lambda,i),(y,\gamma,j)\big) = (d_E(x,y), min\{\lambda,\gamma\}, max\{i,j\}) \tag{3.29}$$

d_E denotes the Euclidean distance, (T, d_{IFE}) is a strong intuitionistic fuzzy metric space and T is the set of intuitionistic fuzzy points on the intuitionistic fuzzy set T.

Proof: Here we only consider the third property of triangle inequality as the first two properties can be proved in the similar way as shown in Proposition 1 of section 3.4.1.

Suppose $(x,\lambda,i),(y,\gamma,j)\ and\ (z,\rho,l)$ are three arbitrary intuitionistic fuzzy points on T. As stated earlier, (R^n, d_E) is a metric space. Therefore,

$$d_E(x,z) \leq\ d_E(y,z) + d_E(x,y) \tag{3.30}$$

If the above stated inequality (3.30) holds strictly, then condition $(3')$ is obviously satisfied from Definition 1(1) of section 3.4.1. For the other case, where the " = " relation is considered, there must exist some $\lambda \in (0,1]$ such that $y = (1-\lambda)x + \lambda z$. Let $\alpha = min\{\lambda,\rho\}$ and $\beta = max\{i,l\}$. Then we can say that $\{x,z\} \subset T_{\alpha,\beta}$. As, T is an intuitionistic fuzzy linear space, $T_{\alpha,\beta}$ is a linear subspace of R^n. It follows that $y \in T_{\alpha,\beta}$ i.e., $\gamma = \mu_T(y) \geq \alpha = min\{\lambda,\rho\}$. This implies that $min\{\lambda,\rho,\gamma\} = min\{\lambda,\rho\}$. Similarly, $j = \nu_T(y) \leq \beta = max\{i,l\}$. This implies that $max\{i,l,j\} = max\{i,l\}$.

Thus it can be stated that:

$$\begin{aligned} d_{IFE}\big((x,\lambda,i),(z,\rho,l)\big) &= (d_E(x,z), min\{\lambda,\rho\}, max\{i,l\}) \\ &= (d_E(x,y) + d_E(y,z), min\{\lambda,\rho,\gamma\}, max\{i,l,j\}) \\ &= d_{IFE}\big((x,\lambda,i),(y,\gamma,j)\big) + d_{IFE}\big((y,\gamma,j),(z,\rho,l)\big) \end{aligned}$$

3.4.2 Intuitionistic Fuzzy Orienteering Problem using Intuitionistic Fuzzy Points

We apply the above derived formula of distance between two IFPs (Proposition 1 of section 3.4.1) on Orienteering Problem (OP) which can be represented as an undirected graph $G(V, E)$ where E signifies the set of edges and $V = \{v_1, \dots \dots .., v_N\}$ is the set of vertices. Let the two functions viz. distance function on edges be denoted as $d: E \rightarrow \Re^+$ and the function for score on vertices can be stated as $S: V \rightarrow \Re^+$. The goal here is to determine a Hamiltonian Path (P) that satisfies the distance bound (D_{max}), connects the source (v_1), target (v_N) and a subset of vertices (V') of V such that the total collected score is maximized (Vansteenwegen, Souffriau, & Oudheusden, 2011). OP finds several real life applications which include the fields of logistics, home delivery system, disaster management system, tourism industry etc. In each of the stated applications, the parameters involved viz. score and distance cannot be determined precisely. Also, there exists some uncertainty in the position or location of the point of interest. Therefore, some technique is required to tackle this uncertainty. One such method is the use of intuitionistic fuzzy points to model the uncertainty in the position and intuitionistic fuzzy numbers to tackle the imprecise nature of the parameter score.

(a) Algorithm

Input: A given graph G with nodes represented as intuitionistic fuzzy points and score of each node represented as a trapezoidal intuitionistic fuzzy number.

Output: A path (P) that satisfies the distance bound D_{max} and maximizes the total collected score.

Step 1: Calculate the intuitionistic fuzzy distance values (i.e., the weight of each edge denoted as (d_{ij}) using the following formula:

$$d_{IF}\big((x, \lambda, i), (y, \gamma, j)\big) = (d(x, y), min\{\lambda, \gamma\}, max\{i, j\}) \tag{3.31}$$

Where, $d(x, y)$ denotes the Euclidean distance between two nodes.

Step 2: Determine all possible paths connecting the source and target. For each path calculate:

(i) The total distance covered by the path using the following formula: (addition of two intuitionistic fuzzy points)

$$(x, \lambda, i) + (y, \gamma, j) = \left((x + y), min\{\lambda, \gamma\}, max\{i, j\}\right) \tag{3.32}$$

(ii) The total score collected on traversing a path using Eq. 3.2 of section 3.2.1 (I).

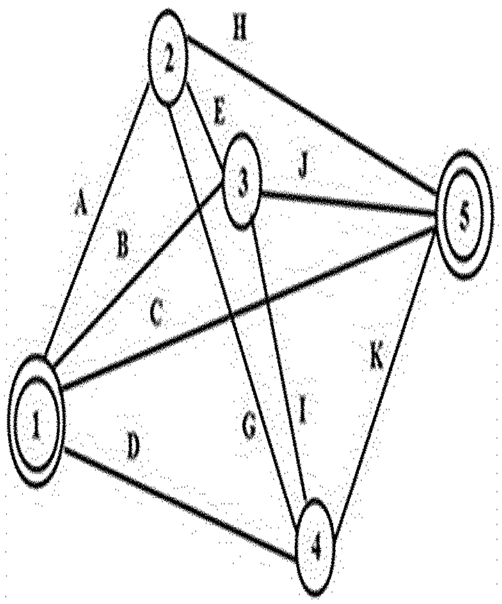
Step 3: Discard those paths from the set of all possible paths that do not satisfy the D_{max} values (using Definition 1(1) of section 3.4.1).

Step 4: Rank the scores of the remaining paths to determine the most desirable path (i.e., the path that satisfies the upper limit on the distance covered and has the best total collected score value using Eqs. 3.13 - 3.15).

Thus, the path with $R = 1$ is considered to be the most desirable path and returned as output. If two paths have the same score values then their rank values are computed on the basis of the total distance covered by those paths (i.e., the path that covers less distance, gets higher ranking).

(b) Illustrative Example

Consider the network in Fig. 3.10.



Node	Label	Intuitionistic Fuzzy Point Values$((x,y),\lambda,i)$	Intuitionistic Fuzzy Score Values $\langle(\mu(x)),(v(x))\rangle$
v_1	1	$((10.5,14.4),0.6,0.3)$	$\langle(1,2,8,9),(0,2,8,10)\rangle$
v_2	2	$((11.2,14.1),0.8,0.2)$	$\langle(8,9,11,12),(6,8,12,14)\rangle$
v_3	3	$((18,15.9),0.5,0.5)$	$\langle(3,5,9,11),(1,4,10,13)\rangle$
v_4	4	$((18.3,13.3),0.7,0.1)$	$\langle(17,20,24,27),(14,18,26,30)\rangle$
v_5	5	$((16.5,9.3),0.9,0.1)$	$\langle(1,2,4,5),(0,1,5,6)\rangle$
$D_{max} = (20,0.8,0.2)$			

Fig. 3.10: Input graph G with number of nodes $(N) = 5$, source $(v_1) = 1$, target $(v_N) = 5$ and the co-ordinate values (intuitionistic fuzzy points) and the score values (trapezoidal intuitionistic fuzzy numbers) of each node

***Step 1*:** The d_{ij} value of each edge is computed using Eq. 3.31 and the values are presented in Table 3.3.

***Step 2*:** All the possible paths are determined and the following values are calculated and presented in Table 3.4.

(i) Total distance covered by the path using Eq. 3.32.

(ii) Total score collected on traversing a path using Eq. 3.2.

Step 3**:** Those paths that do not satisfy the distance bound (D_{max}) are discarded using Definition 1(1) of section 3.4.1 and the remaining paths that form the solution set are stated in Table 3.5.

Step 4**:** The paths obtained after Step 3 are ranked using Eqs. 3.13 - 3.15 to determine the most desirable path and the values are stated in Table 3.6.
Hence, the most desirable path is P_{11} that collects the maximum possible score within the specified distance constraint.

Table 3.3: The d_{ij} value of each edge

Edge	Label	Distance Values of each Edge (d_{ij})
$1-2$	A	$(0.76, 0.6, 0.3)$
$1-3$	B	$(7.64, 0.5, 0.5)$
$1-4$	D	$(7.88, 0.6, 0.3)$
$1-5$	C	$(9.38, 0.6, 0.3)$
$2-3$	E	$(7.03, 0.5, 0.5)$
$2-4$	G	$(7.14, 0.7, 0.2)$
$2-5$	H	$(7.15, 0.8, 0.2)$
$3-4$	I	$(2.62, 0.5, 0.5)$
$3-5$	J	$(6.77, 0.5, 0.5)$
$4-5$	K	$(4.39, 0.7, 0.1)$

Table 3.4: The value of total distance covered and total score collected on traversing each path

Path	Total Distance Covered by the Path	Total Score Collected by the Path
$P_1: 1-5$	$(9.38, 0.6, 0.3)$	$\langle(1,2,8,9), (0,2,8,10)\rangle$
$P_2: 1-2-5$	$(7.91, 0.6, 0.3)$	$\langle(9,11,19,21), (6,10,20,24)\rangle$
$P_3: 1-3-5$	$(14.41, 0.5, 0.5)$	$\langle(4,7,17,20), (1,6,18,23)\rangle$
$P_4: 1-4-5$	$(12.27, 0.6, 0.3)$	$\langle(18,22,32,36), (14,20,34,40)\rangle$
$P_5: 1-2-3-5$	$(14.56, 0.5, 0.5)$	$\langle(12,16,28,32), (7,14,30,37)\rangle$
$P_6: 1-3-2-5$	$(21.82, 0.5, 0.5)$	$\langle(12,16,28,32), (7,14,30,37)\rangle$
$P_7: 1-2-4-5$	$(12.29, 0.6, 0.3)$	$\langle(26,31,43,48), (20,28,46,54)\rangle$
$P_8: 1-4-2-5$	$(22.17, 0.6, 0.3)$	$\langle(26,31,43,48), (20,28,46,54)\rangle$
$P_9: 1-3-4-5$	$(14.65, 0.5, 0.5)$	$\langle(21,27,41,47), (15,24,44,53)\rangle$
$P_{10}: 1-4-3-5$	$(17.27, 0.5, 0.5)$	$\langle(21,27,41,47), (15,24,44,53)\rangle$
$P_{11}: 1-2-3-4-5$	$(14.8, 0.5, 0.5)$	$\langle(29,36,52,59), (21,32,56,67)\rangle$
$P_{12}: 1-2-4-3-5$	$(17.29, 0.5, 0.5)$	$\langle(29,36,52,59), (21,32,56,67)\rangle$
$P_{13}: 1-3-4-2-5$	$(24.55, 0.5, 0.5)$	$\langle(29,36,52,59), (21,32,56,67)\rangle$
$P_{14}: 1-4-3-2-5$	$(24.68, 0.5, 0.5)$	$\langle(29,36,52,59), (21,32,56,67)\rangle$
$P_{15}: 1-4-2-3-5$	$(28.82, 0.5, 0.5)$	$\langle(29,36,52,59), (21,32,56,67)\rangle$
$P_{16}: 1-3-2-4-5$	$(26.2, 0.5, 0.5)$	$\langle(29,36,52,59), (21,32,56,67)\rangle$

Table 3.5: The solution set after discarding those paths that do not satisfy the distance bound ($\boldsymbol{D_{max}}$)

Path	Total Distance Covered by the Path	Total Score Collected by the Path
$P_1: 1-5$	$(9.38, 0.6, 0.3)$	$\langle(1,2,8,9),(0,2,8,10)\rangle$
$P_2: 1-2-5$	$(7.91, 0.6, 0.3)$	$\langle(9,11,19,21),(6,10,20,24)\rangle$
$P_3: 1-3-5$	$(14.41,0.5,0.5)$	$\langle(4,7,17,20),(1,6,18,23)\rangle$
$P_4: 1-4-5$	$(12.27,0.6,0.3)$	$\langle(18,22,32,36),(14,20,34,40)\rangle$
$P_5: 1-2-3-5$	$(14.56,0.5,0.5)$	$\langle(12,16,28,32),(7,14,30,37)\rangle$
$P_7: 1-2-4-5$	$(12.29,0.6,0.3)$	$\langle(26,31,43,48),(20,28,46,54)\rangle$
$P_9: 1-3-4-5$	$(14.65,0.5,0.5)$	$\langle(21,27,41,47),(15,24,44,53)\rangle$
$P_{10}: 1-4-3-5$	$(17.27,0.5,0.5)$	$\langle(21,27,41,47),(15,24,44,53)\rangle$
$P_{11}: 1-2-3-4-5$	$(14.8,0.5,0.5)$	$\langle(29,36,52,59),(21,32,56,67)\rangle$
$P_{12}: 1-2-4-3-5$	$(17.29,0.5,0.5)$	$\langle(29,36,52,59),(21,32,56,67)\rangle$

Table 3.6: Ranks assigned to the paths to determine the most desirable path

Path	Total Distance Covered by the Path	CoC Values for Score	Rank
$P_1: 1-5$	$(9.38, 0.6, 0.3)$	4.417	10
$P_2: 1-2-5$	$(7.91, 0.6, 0.3)$	11.51	8
$P_3: 1-3-5$	$(14.41,0.5,0.5)$	9.846	9
$P_4: 1-4-5$	$(12.27,0.6,0.3)$	20.006	6
$P_5: 1-2-3-5$	$(14.56,0.5,0.5)$	17.396	7
$P_7: 1-2-4-5$	$(12.29,0.6,0.3)$	27.171	3
$P_9: 1-3-4-5$	$(14.65,0.5,0.5)$	25.504	4
$P_{10}: 1-4-3-5$	$(17.27,0.5,0.5)$	25.504	5
$\boldsymbol{P_{11}: 1-2-3-4-5}$	$(14.8,0.5,0.5)$	32.670	**1**
$P_{12}: 1-2-4-3-5$	$(17.29,0.5,0.5)$	32.670	2

3.5 Conclusion

In this chapter, we have modified the PDA algorithm suggested by Chen *et al*. for CSPP to deal with the intuitionistic fuzzy environment. The algorithm suggested for CIFSPP is capable of tackling the uncertainty involved in the parameters and provides a better picture of the real world. Also the problem deals with multiple constraints, namely cost and delay. We represent cost using TIFN and the other parameter viz. delay remains crisp. We introduced a centroid method of ranking for TIFNs which is different from the techniques available in the literature as it uses eight variable representation for TIFN instead of the already existing six variable representation and provides a more generalized representation than the existing methods. Therefore it is a better method to manage the imprecise nature of the parameters involved. Another method of ranking called the centroid of centroids (CoC) has also been proposed for the TIFN. Also, we have introduced the concept of Quasi-Gaussian intuitionistic fuzzy number and tackled the uncertainty that exists in case of practical applications like the CIFSPP by representing one of the parameters viz. cost as a QGIFN while the other parameter that is delay, remains crisp. Also, a centroid based ranking method has been proposed to rank the QGIFNs and determine the cheapest feasible path for CIFSPP.

In this chapter, we have also considered the orienteering problem which is an NP-Hard problem and formulated the intuitionistic fuzzy version of this problem accounting for uncertainty in the real life areas where this problem finds its application like the tourism industry, logistics etc. The problem has been stated as a fuzzy integer program with fuzzy goals and crisp constraints and a fuzzy optimization technique for solving IFOP has been presented. The method suggested here considers the hesitancy, aspiration levels, degree of acceptability and satisfaction of the decision maker, thus providing latitude to the solution process. The generated solution is capable of tackling the uncertainty and vagueness involved in the two parameters score and time. To deal with larger instances efficiently, we presented the work-depth analysis of IFOP and showed that the algorithm is work-preserving and thus can be efficiently implemented on a multiprocessor model like the PRAM.

Here, a new definition for intuitionistic fuzzy metric space using the idea of intuitionistic fuzzy point and intuitionistic fuzzy scalars has been proposed. The concept of IFP and the distance measure thus proposed has been applied for the first time on the intuitionistic fuzzy orienteering problem to model the uncertainty present in the position or location and the two parameters involved viz. distance and score. Also, with the help of an illustrative example, we suggested a method to solve IFOP. In future, some heuristic or meta-heuristic algorithm like the ant colony optimization algorithm, firefly algorithm, flower pollination algorithm etc. can be used to replace the exhaustive search method and apply the proposed technique for larger graphs.