Proc. of the Second Intl. Conf. on Advances in Computing, Electronics and Electrical Technology - CEET 2014.

ISBN: 978-1-63248-034-7 doi: 10.15224/ 978-1-63248-034-7-25

A New Algorithm for Solving Fuzzy Constrained Shortest Path Problem using Intuitionistic Fuzzy Numbers

Madhushi Verma and K. K. Shukla

Abstract— **Constrained shortest path problem (CSPP) is an NP-Complete problem where the goal is to determine the cheapest path bounded by a given delay constraint. This problem finds application in several fields like television and transportation networks, ATM circuit routing, multimedia applications etc. In these kinds of applications it is important to provide quality of service (QoS). Therefore, it is necessary to model the uncertainty involved in the parameters like cost, delay, time etc. The best technique to deal with the imprecise nature of the parameters is to represent them using fuzzy numbers. We propose a solution for the intuitionistic fuzzy version of the problem i.e. constrained intuitionistic fuzzy shortest path problem (CIFSPP) where one of the parameter i.e. cost is represented as a trapezoidal intuitionistic fuzzy number (TIFN) and the other parameter i.e. delay is modelled using real numbers. In this paper, we prefer intuitionistic fuzzy sets (IFS) over ordinary fuzzy set because using IFS both membership and non-membership as well as hesitancy degrees can be represented. Therefore, IFS provides 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 fuzzy representation is ranking of the parameters since fuzzy numbers cannot be ordered naturally. We introduce a centroid based technique of ranking that is different from the existing methods as it uses an eight variable representation for TIFN and is a more generalized method of representing TIFNs. Therefore, it is more appropriate for problems like CIFSPP. An algorithm based on the TIFN ranking method is given and experimental results presented by applying it on large random graphs generated using a power law random graph generator *gengraph-win*. Also, we verify experimentally that the parameters in CIFSPP show the same behaviour as in the crisp version of the algorithm when the delay constraint is relaxed sufficiently.**

Keywords—centroid method, constrained shortest path problem, ranking, trapezoidal intuitionistic fuzzy number.

I. Introduction

Shortest path problem (SPP) is a fundamental and a well-studied problem and lot of algorithms exist in the literature that can solve SPP in polynomial time. It is a combinatorial optimization problem that can be modelled using a graph with real numbers assigned to the edges as weights and the objective of this problem is to determine the cheapest path connecting a pair of source and target. Adding a few more constraints to this objective makes the problem intractable and is represented as constrained shortest path problem (CSPP) [1]. CSPP is an NP-Complete problem and finds application in several fields like transportation networks, cable television networks, computer networks, ATM circuit routing and several multimedia applications like remote diagnosis, web broadcasting and teleconferencing etc. [2], [3].

In 1966, the first algorithm to solve CSPP was proposed by Joksh [4]. As CSPP is an NP-Complete problem, most of the techniques suggested to solve it are either heuristic or approximation algorithms. A few simple and fast heuristics were presented by Luo et al, Ravindran et al etc. [5], [6]. A heuristic with the same complexity as that of the Dijkstra's algorithm was proposed by Korkmaz and Krunz [7]. Most of the methods available in the literature to tackle CSPP, utilized either of the strategies i.e. Lagrangian Relaxation (LR) or Dynamic Programming (DP). Xue proposed a solution for the delay constrained least cost routing problem using the LR strategy [8]. Few approximation algorithms were presented by Sahni and Ibarra et al. [9], [10]. Warburton suggested the first fully polynomial time approximation algorithm for acyclic networks [11]. Later Hassin improved these results and proposed two fully polynomial time approximation schemes (FPTAS) for acyclic networks incorporating the techniques of DP and interval partitioning [12]. For general networks, a strong polynomial time approximation scheme was introduced by Lorenz and Raz [13].

As stated above, CSPP can be applied in several real life situations and the major obstacle in these applications 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. The concept of fuzzy set was introduced by Zadeh in 1965 [14]. When the domain of discourse for a fuzzy set is the real line R, then it is called a fuzzy number. There are several types of fuzzy numbers like the Gaussian fuzzy number,

Madhushi Verma

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

K. K. Shukla
Department of Computer Science and Engineering, IIT (BHU), Varanasi. India

Exponential fuzzy number, quadratic fuzzy number, triangular fuzzy number, trapezoidal fuzzy number etc. [15]. In most of the engineering applications, trapezoidal fuzzy numbers are used as they are simple to represent, easy to understand and have a linear membership function so arithmetic computations can be performed easily [16]. The ordinary fuzzy numbers can be represented through membership functions but there are several practical situations which cannot be modelled through membership functions only. For example, a travelling salesman has to visit a set of cities in order to maximize his sales and has some knowledge about the cities where maximum sales can take place but he does not have enough resources like time or fuel etc. to visit all the cities. In such a scenario, the salesman utilizes his knowledge about the sales to achieve the goal within the specified constraint. Let C denote the set of cities to be visited and $m \in C$ denote the city visited by the salesman, then the membership degree of the cities visited by the salesman can be represented as $\mu(m)$. However, if the requirement is to determine the cities that could not be visited by the salesman, then ordinary fuzzy set is not sufficient to model the situation. In such a case, an extension of the fuzzy set called the intuitionistic fuzzy set (IFS) introduced by Atanassov is required which considers both the degree of belongingness and non-belongingness [17]. So, the degree of non-membership can be evaluated as $\nu(m) = 1 - \mu(m)$. Also, there can be a situation where the salesman visits a city but is not able to sell his product due to some reason like shop being closed or unavailability of the customer. This situation can be modelled using hesitancy degree which can be represented as $\pi(m) = 1 - \left(\mu(m) + \nu(m)\right)$. Few other advantages of IFS include: (1) geometrical interpretation is possible for IFS, therefore the prevailing uncertainty can be tackled in a much better way and (2) operators like modal operators that help in providing a detailed estimation of the information available can only be defined for IFS and not for ordinary fuzzy sets [18].

In this paper, we use trapezoidal intuitionistic fuzzy number (TIFN) to represent one of the parameters involved in CSPP i.e. cost and the other parameter i.e. delay remains crisp. We propose a ranking method for TIFN and modify the CSPP algorithm suggested by Chen et al to deal with the fuzzy environment and provide a solution for the constrained intuitionistic fuzzy shortest path problem (CIFSPP).

II. Preliminaries

A. *Problem Definition*

Any practical network can be modelled through a graph denoted as an ordered pair $G(V, E)$ where V and E represent the set of vertices and set of edges respectively such that $|V| = n \; and \; |E| = m$. An edge belonging to E can be stated as (u, v). The cost and delay values associated with each edge can be specified as $c(u, v)$ and $d(u, v)$ respectively and the same for a path can be stated as $c(P)$ and $d(P)$ respectively. To compute the cost and delay values of a path, the following equation can be used:

$$d(P) = \sum_{(u,v)\in P} d(u, v) \quad (1)$$

$$c(P) = \sum_{(u,v)\in P} c(u, v) \quad (2)$$

The longest path existing in the network is denoted as L and $l(P)$ represents the length (number of hops) of P. In CSPP (as suggested by Chen et al.), a path is feasible if it fulfils the delay constraint $d(P) \leq r$. The goal is to find the cheapest feasible path connecting the source $(s)$and target (t) i.e. the one that has the minimum cost $c\left(P_{s,t}\right)$ among all the feasible paths that satisfies the delay constraint $d(P) \leq r$ [1].

In CIFSPP, the task is to determine a path that is cheapest and also obeys the delay constraint for a graph with intuitionistic fuzzy cost and crisp delay associated with each edge of G. Since the $c(u, v)$ are intuitionistic fuzzy numbers, to calculate the $c(P)$ value (5) is used. Finally to conclude with the cheapest feasible path the ranking method specified in section II (E) is used.

B. *Path Delay Discretization (PDA)*

CSPP is a well-known NP-Complete problem and various methods have been suggested to deal with the problem. One strategy to tackle CSPP is to reduce it to a polynomial time solvable one by using the technique of discretization. However, the effectiveness of this technique can be measured by considering the amount of error induced during discretization. Chen et al suggested two methods of discretization to provide for more efficient network functions and optimum utilization of limited resources. These techniques include (a) path delay discretization (PDA) and (b) randomized discretization. To solve CIFSPP, we prefer PDA over randomized discretization because the problem of error accumulation is absent in case of PDA as discretization is performed on the path delays using interval partitioning.

Let $d(P)$ represent the path delay, r denote the delay constraint and λ be the integer that bounds the delay constraint. Thus, discretized delay can be computed using the following equation [1]:

$$d'(P) = \left\lfloor \frac{d(P)}{r}\lambda \right\rfloor \quad (3)$$

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

C. *Trapezoidal Intuitionistic Fuzzy Number (TIFN)*

Atanassov proposed the concept of intuitionistic fuzzy set as an extension of fuzzy sets. In IFS two values are associated with every element belonging to the universal set. One signifies the degree of membership and the other denotes the degree of non-membership. Both the values lie within the [0, 1] unit interval. An IFS X in U can be defined as $X = \{\langle x, \mu_X(x), \nu_X(x)\rangle : x \in U\}$ with a constraint that $0 \leq \mu_X(x) + \nu_X(x) \leq 1 \; \forall x \in U$ where U is the universe of discourse. Another term that can be associated with every element $x \in U$ is the hesitancy degree which can be represented as $\pi_X(x) = 1 - \mu_X(x) - \nu_X(x)$ such that $0 \leq \pi_X(x) \leq 1$ [19].

When the universe of discourse U for an IFS is the real line i.e. $U = \mathcal{R}$, then it is called an intuitionistic fuzzy number

(IFN). An IFN can be denoted as $X = \{\langle x, \mu_X(x), \nu_X(x)\rangle : x \in \mathcal{R}\}$ and possess the following properties [20]:

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

(b) For at least two points $x_1\ and\ x_2$ in U, $\mu_X(x_1) = 1$ and $\nu_X(x_2) = 1$.

(c) The membership function (μ_X) is upper semicontinuous and the non-membership function (ν_X) is lower semicontinuous.

In this paper, among the two parameters involved in CIFSPP i.e. cost and delay, cost is represented as a 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 [19]

$$\mu_A(x) = \begin{cases} 0 & if\ x < a \\ L_A(x) & if\ a \le x < b \\ 1 & if\ b \le x \le c \\ M_A(x) & if\ c < x \le d \\ 0 & if\ d < x \end{cases} \qquad (4a)$$

$$\nu_A(x) = \begin{cases} 1 & if\ x < e \\ N_A(x) & if\ e \le x < f \\ 0 & if\ f \le x \le g \\ K_A(x) & if\ g < x \le h \\ 1 & if\ h < x \end{cases} \qquad (4b)$$

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

$K_A(x) = \dfrac{x-g}{h-g}$

.

D. *Arithmetic operations for TIFN*

In problems like 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 is discussed in detail in the next section. Here we state the equation for addition of TIFNs [19].

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

$$A_1 + A_2 = \langle (a_1,b_1,c_1,d_1),(e_1,f_1,g_1,h_1)\rangle + \langle (a_2,b_2,c_2,d_2),(e_2,f_2,g_2,h_2)\rangle = \langle (a_1+a_2, b_1+b_2, c_1+c_2, d_1+d_2),(e_1+e_2, f_1+f_2, g_1+g_2, h_1+h_2)\rangle \qquad (5)$$

E. *Ranking Method*

To rank the TIFNs we introduce a centroid method of ranking in this paper. Several other methods of ranking are also available in the literature and in [21], 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 (4a), (4b) and Fig. 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 $z: \mathbb{R} \to [0,1]$ such that

$$z(x) = \frac{(\mu-\nu)(x)+1}{2} \qquad (6)$$

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

$$\text{Now, the centroid } (C) = \frac{\int_e^h z(x)x\,dx}{\int_e^h z(x)dx} \qquad (7)$$

Using (6), we can say that

$$(C) = \frac{\int_e^h \frac{(\mu-\nu)(x)+1}{2} x\,dx}{\int_e^h \frac{(\mu-\nu)(x)+1}{2} dx} \qquad (8)$$

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

From (4a) and (4b) we can say that

$$\int_e^h \mu(x)\,x\,dx = \int_a^b \left(\frac{x-a}{b-a}\right) x\,dx + \int_b^c (1)\,x\,dx + \int_c^d \left(\frac{x-d}{c-d}\right) x\,dx = \frac{(c^2+d^2+cd)-(a^2+b^2+ab)}{6} \qquad (9)$$

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

$$\int_e^h [1-\nu(x)]\,x\,dx = \frac{(g^2+h^2+gh)-(e^2+f^2+ef)}{6} \qquad (10)$$

Therefore,

$$\int_e^h z(x)x\,dx = \frac{(c^2+d^2+cd)+(g^2+h^2+gh)-(a^2+b^2+ab)-(e^2+f^2+ef)}{12} \qquad (11)$$

Similarly using (4a), (4b) and integration, we evaluate the function in the denominator of (7) and get the following equations:

$$\int_e^h \mu(x)\,dx = \frac{(c+d)+(a+b)}{2} \qquad (12)$$

$$\int_e^h [1-\nu(x)]\,dx = \frac{(g+h)+(e+f)}{2} \qquad (13)$$

Therefore,

$$\int_e^h z(x)\,dx = \frac{(c+d+g+h)-(a+b+e+f)}{4} \qquad (14)$$

The formula for calculating the centroid of a TIFN is derived using (7), (11) and (14) and is presented below:

$$(C) = \frac{[(c^2+d^2+cd)+(g^2+h^2+gh)-(a^2+b^2+ab)-(e^2+f^2+ef)]}{3[(c+d+g+h)-(a+b+e+f)]} \qquad (15)$$

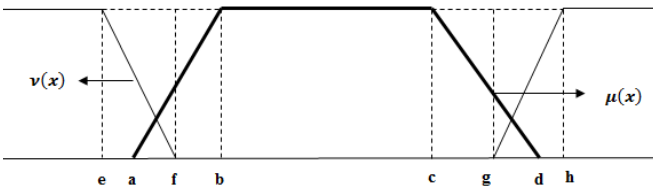


Figure 1. Trapezoidal intuitionistic fuzzy number (TIFN)

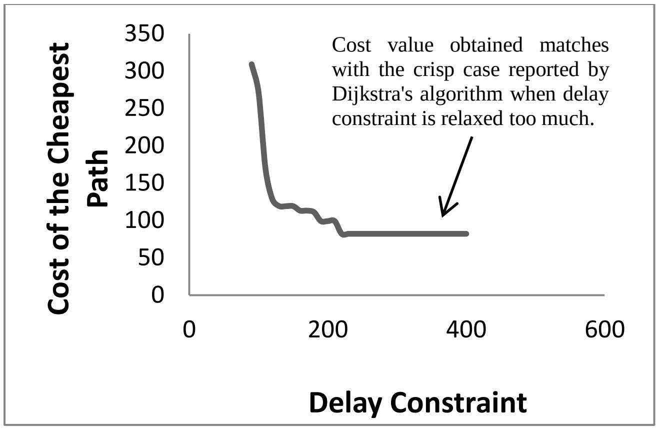
III. Algorithm

```
$\boldsymbol{Get\_abcd}(\ \boldsymbol{alpha1}, \boldsymbol{stretch1}, \boldsymbol{cost})$
$a = cost - stretch1$
$b = a + alpha1$
$d = cost + stretch1$
$c = d - alpha1$
$\boldsymbol{Get\_efgh}(\ \boldsymbol{alpha2}, \boldsymbol{stretch2}, \boldsymbol{cost})$
$e = cost - stretch2$
$f = e + alpha2$
$h = cost + stretch2$
$g = h - alpha2$
$\boldsymbol{Initialize}\ (\boldsymbol{V}, \boldsymbol{s}, \boldsymbol{\lambda})$
$\boldsymbol{for}\ each\ vertex\ \boldsymbol{v}\ \boldsymbol{\epsilon}\ \boldsymbol{V}, each\ \boldsymbol{i}\ \boldsymbol{\epsilon}\ [\boldsymbol{0}, \dots \dots, \boldsymbol{\lambda}]$
$\boldsymbol{Get\_abcd}(\ \boldsymbol{alpha1}, \boldsymbol{stretch1}, \boldsymbol{cost})$,
    $\boldsymbol{Get\_efgh}(\ \boldsymbol{alpha2}, \boldsymbol{stretch2}, \boldsymbol{cost})$
$w[v,i] \coloneqq\ \infty, \pi[v,i] \coloneqq NIL, z[v,i] \coloneqq\ \infty$
    $w[s,0] \coloneqq 0, z[s,i] \coloneqq 0$
  $\boldsymbol{end\ for}$
$\boldsymbol{Relax\_FPDA}(\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{i}, \boldsymbol{\lambda})$
  $i' \coloneqq\ floor\left(\frac{z[u,i]+\ d(u,v)}{r}\lambda\right)$
  $\boldsymbol{Get\_abcd}(\ \boldsymbol{alpha1}, \boldsymbol{stretch1}, \boldsymbol{cost})$,
  $\boldsymbol{Get\_efgh}(\ \boldsymbol{alpha2}, \boldsymbol{stretch2}, \boldsymbol{cost})$
  $\boldsymbol{if}\ i' \leq\ \lambda\ and\ w[v,i'] > w[u,i] + c(u,v)$
// Compare using Equation 15 of section
II(E).
          $w[v,i'] \coloneqq w[u,i] + c(u,v)$
           $\pi[v,i'] \coloneqq u$
           $z[v,i'] \coloneqq min\{z[v,i'],\ z[u,i] + d(u,v)\}$
  $\boldsymbol{end\ if}$
$\boldsymbol{FPDA\_Dijkstra}(\boldsymbol{G}, \boldsymbol{s}, \boldsymbol{\lambda})$
$\boldsymbol{Initialize}\ (\boldsymbol{V}, \boldsymbol{s}, \boldsymbol{\lambda})$
  $\boldsymbol{for}$ i = 0 to $\lambda$
       $Q \coloneqq V$
       $\boldsymbol{while}\ Q \neq \varphi$
            $u \coloneqq Extract\_Min(Q)$
            $\boldsymbol{if}\ w[u,i] = \infty$
                  $\boldsymbol{break}\ out\ of\ the\ \boldsymbol{while}\ loop$
            $\boldsymbol{end\ if}$
  $Q \coloneqq Q - \{\ u\ \}$
            $\boldsymbol{for}\ every\ adjacent\ node\ \boldsymbol{v}\ of\ \boldsymbol{u}$
                      $\boldsymbol{Relax\_FPDA}(\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{i}, \boldsymbol{\lambda})$
$\boldsymbol{end\ for}$
          $\boldsymbol{end\ while}$
  $\boldsymbol{end\ for}$
$\boldsymbol{FPDA}(\boldsymbol{G}, \boldsymbol{s})$
$\lambda \coloneqq \lambda_0$
  $\boldsymbol{do}$
$\lambda \coloneqq 2\lambda$
          $\boldsymbol{FPDA\_Dijkstra}(\boldsymbol{G}, \boldsymbol{s}, \boldsymbol{\lambda})$
$\boldsymbol{while}\ \exists v\ \epsilon\ d(P^v) > (1+\epsilon)r$ // where $P^v$ is
the path with $min\{w[v,i] |\ i\ \epsilon\ [0 \dots .. \lambda]\}$
```

This algorithm is an extension of the Path Delay Discretization Algorithm (PDA) suggested by Chen et al [1]. 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. For the detailed explanation of the algorithm, the readers are suggested to refer to [22]. The analysis of the solution generated from the intuitionistic fuzzy version of the above stated algorithm is presented in the next section. Since in the intuitionistic fuzzy version, the number of arithmetic operations increase by a constant factor, the time complexity of the algorithm remains $O((m + n \log n)L/\varepsilon)$, same as stated by Chen et al in [1].

IV. 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 $n = 200, alpha = 2.5, min = 10, max = 20, z = 12, stretch1 = 2, alpha1 = 1, stretch2 = 3.5, alpha2 = 1,$ $\text{source} = 15$ and $\text{target} = 95$. 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. 2



Cost value obtained matches with the crisp case reported by Dijkstra's algorithm when delay constraint is relaxed too much.

Figure 2. 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*.

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 by setting the width of the intuitionistic fuzzy number representing the cost of the final path equal to zero.

V. Conclusion

In this paper, 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 i.e. cost and delay. We represent cost using TIFN and the other parameter i.e. 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.

Acknowledgment

Programming efforts by Sanchit Mehrotra and Uttam Kanodia are gratefully acknowledged. The first author would like to acknowledge the financial support by IIT (BHU) in terms of teaching assistantship.

References

[1] S. Chen, M. Song, and S. Sahni, “Two Techniques for Fast Computation of Constrained Shortest Paths,” IEEE/ACM Transactions on Networking, vol. 16, no. 1, pp. 105-115, February 2008.

[2] Y. Xiao, K. Thulasiraman, G. Xue, and A. Juttner, “The Constrained Shortest Path Problem: Algorithmic Approaches and an Algebraic Study withGeneralization”. http://www.cs.elte.hu/~alpar/publications/jour/AKCE_October_25.pdf.

[3] Y. Dou, L. Zhu, and H. S. Wang, “Solving the fuzzy shortest path problem using multi-criteria decision method based on vague similarity measure,” Applied Soft Computing, vol. 12, pp. 1621-1631, 2012.

[4] H. Joksch, “The shortest route problem with constraints,” Journal of Mathematical Analysis and Applications, vol. 14, no. 1, pp. 191-197, 1966.

[5] G. Luo, K. Huang, J. Wang, C. Hobbs, and E. Munter, “Multi-QoS Constraints Based Routing for IP and ATM Networks,” presented at the IEEE Workshop on QoS Support for Real-Time Internet Applications, Vancouver Canada, June 1, 1999.

[6] R. Ravindran, K.Thulasiraman, A. Das, K. Huang, G. Luo, and G. Xue, “Quality of Services Routing: Heuristics and Approximation Schemes with a Comparative Evaluation,” in Proc. of ISCAS, 2002.

[7] T. Korkmaz, and M. Krunz, “Multi-constrained optimal path selection,” in Proc. of IEEE INFOCOM, pp. 834–843, 2001.

[8] G. Xue, “Primal-dual algorithms for computing weight-constrained shortest paths and weight-constrained minimum spanning trees,” in Proc. of IEEE IPCCC’00, pp. 71–277, Feb. 2000.

[9] O. Ibarra, and C. Kim, “Fast Approximation Algorithms for the Knapsack and Sum of Subsets Problems,” Journal of the Association for Computing Machinery, vol. 22, pp. 463-468, October 1975.

[10] S. Sahni, “General Techniques for Combinatorial Approximation,” Operations Research, vol. 25, pp. 920-936, 1977.

[11] A. Warburton, “Approximation of pareto optima in multiple-objective shortest path problems,” Operations Research, vol. 35, pp. 70-79, 1987.

[12] R. Hassin, “Approximation Schemes for the Restricted Shortest Path Problem,” Mathematics of Operation Research, vol. 17, no.1, pp. 36-42, 1992.

[13] D. Lorenz, and D. Raz, “A Simple Efficient Approximation Scheme for the Restricted Shortest Paths Problem,” Operations Research Letters, vol. 28, pp. 213-219, 2001.

[14] L. A. Zadeh, “Fuzzy sets,” Inf. Control, vol. 8, pp. 338–353, 1965.

[15] M. Hanss, Applied Fuzzy Arithmetic: An Introduction with Engineering Applications. Netherlands: Springer Berlin Heidelberg New York, 2005, ch. 2, pp. 45-78.

[16] A. Bansal, “ Trapezoidal Fuzzy Numbers (a, b, c, d): Arithmetic Behavior,” International Journal of Physical and Mathematical Sciences, pp. 39-44, 2011.

[17] K. Atanassov, “ Intuitionistic fuzzy sets,” Fuzzy Sets and Systems, vol. 20, pp. 87–96, 1986.

[18] K. Atanassov, “Intuitionistic fuzzy sets: past, present and future,” in Proc. of 3rd Conference of the European Society for Fuzzy Logic and Technology, Zittau, Germany, September, 2003.

[19] J. Ye, “Expected value method for intuitionistic trapezoidal fuzzy multicriteria decision-making problems,” Expert Systems with Applications, vol. 38, pp. 11730-11734, 2011.

[20] P. Grzegorzewski, “Distances and orderings in a family of intuitionistic fuzzy numbers,” in Proc. of EUSFLAT, 2003, pp. 223-227, Germany, September 10-12, 2003.

[21] A.Varghese, and S. Kuriakose, “Centroid of an intuitionistic fuzzy number, ” Notes on Intuitionistic Fuzzy Sets, vol. 18, no. 1, pp. 19-24, 2012,

[22] M. Verma, and K. K. Shukla, “Fuzzy Constrained Shortest Path Algorithm using Circumcenter of Centroids,” in Proc. of IEEE IACC’13, pp. 657-660 Ghaziabad, February 22-23, 2013.