Fuzzy Shortest Path Algorithm Based on Comparative Relation

V.T.T Huyen[1], N.T. Luan[2], and L.M. Tuan[3]

[1] University of Economic and Technical Industries, Namdinh, Vietnam
[2, 3] Thang Long University, Hanoi, Vietnam

Summary
The problem of finding the shortest path is one problem attracted attention of many researchers because it is widely used in many fields such as communication, routing and transportation. In the traditional problem, the length of the edge is represented by the exact value and finding the shortest path has solved by Dijkstra's algorithm. But in fact, the length of the edge is usually expressed by the uncertain value and then we have the model of the Fuzzy Shortest Path Problem. In this paper, we focus on developing an algorithm to find the shortest path in which the weights of the edges are represented by triangular fuzzy numbers. The mathematical basic of the algorithm is based on the concept of Defined Strict Comparative Relation Function on the set of Triangular Fuzzy Numbers.
Key words:
Fuzzy shortest path, comparative relation, Dijkstra's algorithm.

1. Introduction

The single-source problem of finding a shortest path from a source vertex to a destination vertex is one of the fundamental problems in graph theory and applied in many practical applications. Suppose there is a shipping company needs to ship goods from city S to city D. With a map in hand, the driver has to determine the route from S to D through the cities A, B, and C ... so that the total cost is minimal. Typically, the length (weight) of the route is usually measured by time; cost; … In fact, this length is not always calculated correctly. For example: time of travelling from A to B is about 200 minutes. However, in positive conditions such as the good weather, the good health, this time can be reduced to 120 minutes. And in the negative conditions for example: because of the rain, the road is repairing, traffic jam…time to complete this distance can be up to 300 minutes or more. Clearly, in such cases, we must use fuzzy values to represent weights.
The fuzzy shortest path problem (FSPP) was first analysed by Dubois and Prade in 1980 and solved by Floyd's and Ford's algorithm. Although it is possible to calculate the length of the shortest path but this cannot correspond in any way to any reality.
From 1980 to now, there have been many researches focused on how to solve the FSPP problem. These methods mentioned in references include:

(1): Using the properties of the network flow for converting FSPP to the multiple purpose linear programming problems, Jing-Rung Yu and Tzu-Hao Wei presented this method in [2].

(2): Find all possible paths from the source vertex s to vertex t then specify a length 'minimal' and find the path length similar to the length 'minimal' best. Sujatha and Elizabeth presented this method in [3]. Tzung Jung Nan Yuan Chuang and Kung [1] also built algorithm with the same idea but in the case for the weights are in discrete fuzzy sets

(3): Using 'ranking function' and modifying the labelling Dijkstra's algorithm. A ranking function is defined as a mapping from a fuzzy number set to space R. $g(.): FN \to R$ is called a ranking function if:

$g(\tilde{a}) < g(\tilde{b}) \Rightarrow \tilde{a}$ is less than $\tilde{b}$;

$g(\tilde{a}) > g(\tilde{b}) \Rightarrow \tilde{a}$ is greater than $\tilde{b}$;

$g(\tilde{a}) = g(\tilde{b}) \Rightarrow \tilde{a}$ is equal to $\tilde{b}$.

(4): Developing the own measurements and algorithms, Amit Kumar and Manjot Kaur [4] had quoted the Nayeem and Pal's algorithm presented in 2005 based on the concept of acceptability index to the proposition 'A is inferior to B'.

Overall, in the above methods, method (3) with 'ranking function' is the simplest method. However, as we had analysed in [6], the disadvantages of using 'ranking function' is not found any satisfactory formula $g(\tilde{a}) = g(\tilde{b})$ implies $\tilde{a} = \tilde{b}$ in accordance to the equality definition of fuzzy numbers. This means that, by the 'ranking function' method, the fuzzy shortest paths don't have equal of length which is just having the 'ranking function' equal of value.
To overcome the drawbacks of the 'ranking function' method, in [6] we had introduced a definition of comparative relation of fuzzy numbers. In this paper, we propose a new approach to compare two triangular fuzzy numbers, and on the basic of this notation we solve FSPP problem by applying Dijkstra's algorithm based on a strict comparative relation.
This paper is organized in 5 sections. Section 1 provides a brief introduction about the problem of finding shortest path in a graph with the fuzzy weight and some solutions. Section 2 presents the model and the single-source shortest path problem and Dijkstra's algorithm. Section 3 introduces triangular fuzzy numbers, comparative relation

Manuscript received May 5, 2014
Manuscript revised May 20, 2014

of triangular fuzzy numbers and gives some results about the strict comparative relation. Section 4 presents the model and how to solve single-source shortest path problem for the case, which the weights of the edges are triangular fuzzy numbers. Illustration of algorithm is also presented in section 4 and some conclusions are drawn in section 5.

2. The single-source shortest path problem and Dijkstra's algorithm

2.1 Graph

Given a directed and connected graph $G = \{V, E\}$ where $V = \{1,..,n\}$ is the set of n vertices, and E is the set of edges of the graph. The weight of the edge (i, j) connected from vertex i to vertex j and symbols w_{ij}. A path from vertex u to vertex v of the graph is a sequence of vertices $u, v_1, v_2,.., v_m, v$ such that $u, v, v_1,.., v_m \in V$ and $(u, v_1), (v_1, v_2),..., (v_m, v) \in E$. Length of the path is the sum of the weights of the edges in the path.
The single-source shortest path problem is the problem of finding shortest paths from source vertex s to all other vertices in the graph.

2.2 Dijkstra's algorithm

Dijkstra's algorithm or Labelling algorithm conceived by computer scientist Edsger Dijkstra in 1956 and published in 1959 that solves the single-source shortest path for a graph with non-negative edges path costs. The idea of the algorithm is labelling each vertex of the graph corresponding to the length of the shortest path from source to considering edge.
Denote $d[v]$ is a label of $v \in V$. At the beginning, assign $d[s] = 0$ and $d[v] = \infty$ for all other nodes where ∞ is a very big number lies in range of computer representation. To defined a vertex of the graph that has been reviewed or not, we use an array $u[]$ with $u[v] = false$ means v has not been reviewed.
The algorithm is in Java language [7]:

```
for (int i=0; i<n; i++) {

  // finding minimum label among unvisited vertices
  int v = -1;
  for (int j=0; j<n; j++) {
    if (u[j] == false && (v == -1 || d[j]<d[v])) {
      v = j;
    }
  }

  // set as visited
  u[v]=true;

  for (int j=0; j<n; j++) {
    // if there is exists path between v and j
    if (w[v][j]>0) {
      if (d[v] + w[v][j] <d [j]) {
        d[j] = d[v]+w[v][j];
      }
    }
  }
}
```

Fig. 1 Dijkstra's algorithm

3. Triangular fuzzy numbers and comparative relation on it

Definition 3.1 ([4], [5], [6]): Triangular fuzzy number $\tilde{A}$ is defined by a triplet $(a; \alpha, \beta)$ with the membership function defined as:

$$\mu_{\tilde{A}}(x) = \begin{cases} 1 - \dfrac{a - x}{\alpha} & , \quad a - \alpha < x \le a \\ 1 - \dfrac{x - a}{\beta} & , \quad a < x < a + \beta \\ 0 & , \quad otherwise \end{cases}$$

Let TFN be the set of all Triangular Fuzzy Numbers.

Definition 3.2 ([4],[5],[6])**:** Assuming that both $\tilde{A} = (a; \alpha_{\tilde{A}}, \beta_{\tilde{A}})$ and $\tilde{B} = (b; \alpha_{\tilde{B}}, \beta_{\tilde{B}})$ are triangular fuzzy numbers, the arithmetic operations on triangular fuzzy numbers are as follows:

- $\tilde{A} + \tilde{B} = (a + b; \alpha_{\tilde{A}} + \alpha_{\tilde{B}}, \beta_{\tilde{A}} + \beta_{\tilde{B}})$ (3.1)
- $k\tilde{A} = \begin{cases} (ka; k\alpha_{\tilde{A}}, k\beta_{\tilde{A}}) & , \quad k \ge 0 \\ (ka; k\beta_{\tilde{A}}, k\alpha_{\tilde{A}}) & , \quad k < 0 \end{cases}$ (3.2)

Definition 3.3 ([6]): Two triangular fuzzy numbers $\tilde{A} = (a; \alpha_{\tilde{A}}, \beta_{\tilde{A}})$ and $\tilde{B} = (b; \alpha_{\tilde{B}}, \beta_{\tilde{B}})$ are said to be equivalent if and only if $a = b$ and $\alpha_{\tilde{A}} = \alpha_{\tilde{B}}$ and $\beta_{\tilde{A}} = \beta_{\tilde{B}}$.
We denote $\tilde{A} \underset{\Psi}{<} \tilde{B}$ is form of proposition $\tilde{A}$ less than $\tilde{B}$ according to the definition of relation Ψ .

Definition 3.4 ([6]): The relation Ψ is called a comparative relation if and only if it is a complete ordering relation and is compatible with the ordering relation on the set of real numbers, means that: $\forall a, b \in R$, if $a \Theta b$ then $a \underset{\Psi}{\Theta} b$ where $\Theta \in \{<, \le, =, <>, >, \ge\}$.

Definition 3.5: The comparative relation Γ is called a strict comparative relation if Γ is a comparative relation and satisfies the following conditions:

- If $\tilde{A} \underset{\Gamma}{\le} \tilde{B}$ and $\tilde{C} \underset{\Gamma}{\le} \tilde{D}$ then $\tilde{A} + \tilde{C} \underset{\Gamma}{\le} \tilde{B} + \tilde{D}$;

- If $\tilde{A} \underset{\Gamma}{\le} \tilde{B}$ and $k \ge 0$ then $k\tilde{A} \underset{\Gamma}{\le} k\tilde{B}$.

Definition 3.6: The function *f: TFN×TFN → R* is called Defined Comparative Relation Function (DCRF) on *TFN* if it satisfies the following conditions:

(*i*) $f(\tilde{A},\tilde{A}) \ge 0 \ \forall \tilde{A} \in TFN$;

(*ii*) If $f(\tilde{A},\tilde{B}) \ge 0$ and $f(\tilde{B},\tilde{A}) \ge 0$ then $\tilde{A} = \tilde{B}$;

(*iii*) If $f(\tilde{A},\tilde{B}) \ge 0$ and $f(\tilde{B},\tilde{C}) \ge 0$ then $f(\tilde{A},\tilde{C}) \ge 0$.

Definition 3.7: The DCRF on *TFN* is called Defined Complete Comparative Relation Function (DCCRF) if it satisfies connected conditions:

$\forall \tilde{A},\tilde{B} \in TFN$ either $f(\tilde{A},\tilde{B}) \ge 0$ or $f(\tilde{B},\tilde{A}) \ge 0$.

Definition 3.8: The DCRF on *TFN* is called Defined Strict Comparative Relation Function (DSCRF) if it satisfies the following conditions:

- If $f(\tilde{A},\tilde{B}) \ge 0$ and $f(\tilde{C},\tilde{D}) \ge 0$ then $f(\tilde{A}+\tilde{C},\tilde{B}+\tilde{D}) \ge 0$;
- If $f(\tilde{A},\tilde{B}) \ge 0$ and $k \ge 0$ then $f(k\tilde{A},k\tilde{B}) \ge 0$

Theorem 3.1: Exist the function *f* is DSCRF on *TFN*

Proof: We put

$$f(\tilde{A},\tilde{B}) = \omega \text{sign}(b-a) + \omega_1 \text{sign}(\beta_{\tilde{B}} - \beta_{\tilde{A}}) + \omega_2 \text{sign}(\alpha_{\tilde{B}} - \alpha_{\tilde{A}}) \quad (3.3)$$

where $\omega > \omega_1 > \omega_2 \ge 0$ and $\omega > \omega_2 + \omega_1$ (3.4)

and $\text{sign}(x) = \begin{cases} 1 & , \quad x > 0 \\ -1 & , \quad x < 0 \\ 0 & , \quad x = 0 \end{cases}$

- Firstly, we show that *f* is DCRF.

(i) For every $\forall \tilde{A} \in TFN$ we have $f(\tilde{A},\tilde{A}) = 0$

(ii) If $f(\tilde{A},\tilde{B}) \ge 0$ then by (3.3) implies sign(*b*-*a*) ≥ 0, because if opposite then since $\omega > \omega_2 + \omega_1$ implies $f(\tilde{A},\tilde{B}) < 0$. Consequently, $b \ge a$.

Similarly, from $f(\tilde{B},\tilde{A}) \ge 0$ implies $a \ge b$. And hence *a*=*b*.

Now value of *f* according to formula (3.3) becomes:

$$f(\tilde{A},\tilde{B}) = \omega_1 \text{sign}(\beta_{\tilde{B}} - \beta_{\tilde{A}}) + \omega_2 \text{sign}(\alpha_{\tilde{B}} - \alpha_{\tilde{A}}) \ge 0$$

From there we have also $\beta_{\tilde{B}} - \beta_{\tilde{A}} \ge 0$, because if opposite then since $\omega_1 > \omega_2 \Rightarrow f(\tilde{A},\tilde{B}) < 0$, implies $\beta_{\tilde{B}} \ge \beta_{\tilde{A}}$.

On the other hand, since $f(\tilde{B},\tilde{A}) \ge 0$ we have $\beta_{\tilde{B}} \le \beta_{\tilde{A}}$, therefore $\beta_{\tilde{B}} = \beta_{\tilde{A}}$

Now, *f* is simplified:

$$f(\tilde{A},\tilde{B}) = \omega_2 \text{sign}(\alpha_{\tilde{B}} - \alpha_{\tilde{A}}) \ge 0$$

And this implies, $\alpha_{\tilde{B}} \ge \alpha_{\tilde{A}}$

Likewise, we have $\alpha_{\tilde{B}} \le \alpha_{\tilde{A}}$

Combine two inequalities, we obtain: $\alpha_{\tilde{B}} = \alpha_{\tilde{A}}$.

By the definition 3.3 we have $\tilde{A} \equiv \tilde{B}$.

(iii) If $f(\tilde{A},\tilde{B}) \ge 0$ and $f(\tilde{B},\tilde{C}) \ge 0$ then sign(*b*-*a*) ≥ 0 and from sign(*c*-*b*) ≥ 0 implies *c* ≥ *b* ≥ *a*. There are two cases:

+ Case 1: $c > a \Rightarrow f(\tilde{A},\tilde{C}) \ge 0$

+ Case 2: *c* = *b* = *c*

$$f(\tilde{A},\tilde{B}) = \omega_1 \text{sign}(\beta_{\tilde{B}} - \beta_{\tilde{A}}) + \omega_2 \text{sign}(\alpha_{\tilde{B}} - \alpha_{\tilde{A}}) \ge 0$$

and

$$f(\tilde{B},\tilde{C}) = \omega_1 \text{sign}(\beta_{\tilde{C}} - \beta_{\tilde{B}}) + \omega_2 \text{sign}(\alpha_{\tilde{C}} - \alpha_{\tilde{B}}) \ge 0$$

Therefore, we have

$\beta_{\tilde{B}} - \beta_{\tilde{A}} \ge 0$ and $\beta_{\tilde{C}} - \beta_{\tilde{B}} \ge 0$ or $\beta_{\tilde{C}} \ge \beta_{\tilde{B}} \ge \beta_{\tilde{A}}$.

There are two capabilities:

- Or $\beta_{\tilde{C}} > \beta_{\tilde{A}}$, when

$$f(\tilde{C},\tilde{A}) = \omega_1 \text{sign}(\beta_{\tilde{C}} - \beta_{\tilde{A}}) + \omega_2 \text{sign}(\alpha_{\tilde{B}} - \alpha_{\tilde{A}}) \ge 0$$

- Or $\beta_{\tilde{C}} = \beta_{\tilde{B}} = \beta_{\tilde{A}}$.

By the formula (3.6):

$f(\tilde{A},\tilde{B}) = \omega_2 \text{sign}(\alpha_{\tilde{B}} - \alpha_{\tilde{A}}) \ge 0 \Rightarrow \alpha_{\tilde{B}} - \alpha_{\tilde{A}} \ge 0 \Rightarrow \alpha_{\tilde{B}} \ge \alpha_{\tilde{A}}$ and

$f(\tilde{B},\tilde{C}) = \omega_2 \text{sign}(\alpha_{\tilde{C}} - \alpha_{\tilde{B}}) \ge 0 \Rightarrow \alpha_{\tilde{C}} - \alpha_{\tilde{B}} \ge 0 \Rightarrow \alpha_{\tilde{C}} \ge \alpha_{\tilde{B}}$

This implies that: $f(\tilde{A},\tilde{C}) = \omega_2 \text{sign}(\alpha_{\tilde{C}} - \alpha_{\tilde{A}}) \ge 0$.

- Next, we have to prove *f* is DCRF. Definitely:

$$f(\tilde{A},\tilde{B}) + f(\tilde{B},\tilde{A}) = \omega\ (sign\ (b-a) + sign\ (a-b)) + \omega_1 (sign\ (\beta_{\tilde{B}} - \beta_{\tilde{A}}) + sign\ (\beta_{\tilde{A}} - \beta_{\tilde{B}})) + \omega_2 (sign\ (\alpha_{\tilde{B}} - \alpha_{\tilde{A}}) + sign\ (\alpha_{\tilde{A}} - \alpha_{\tilde{B}})) = 0$$

When either $f(\tilde{A},\tilde{B}) \ge 0$ or $f(\tilde{B},\tilde{A}) \ge 0$.

- Finally, we show that *f* is DSCRF, clearly, in (3.3), if $f(\tilde{A},\tilde{B}) \ge 0$ and $f(\tilde{C},\tilde{D}) \ge 0$ then these are following changes:

+ $[(b > a) \wedge (d > c)] \Rightarrow b+d > a+c$;

+ $[(b > a) \wedge (d = c)] \Rightarrow b+d > a+c$;

+ $[(b = a) \wedge (d > c)] \Rightarrow b+d > a+c$;

Therefore $f(\tilde{A}+\tilde{C},\tilde{B}+\tilde{D}) \ge 0$;

+ $[(b = a) \wedge (d = c)]$ we have

$$f(\tilde{A},\tilde{B}) = \omega_1 \text{sign}(\beta_{\tilde{B}} - \beta_{\tilde{A}}) + \omega_2 \text{sign}(\alpha_{\tilde{B}} - \alpha_{\tilde{A}}) \ge 0$$

And $f(\tilde{C},\tilde{D}) = \omega_1 \text{sign}(\beta_{\tilde{D}} - \beta_{\tilde{C}}) + \omega_2 \text{sign}(\alpha_{\tilde{D}} - \alpha_{\tilde{C}}) \ge 0$,

Similarly:

- $[(\beta_{\tilde{B}} > \beta_{\tilde{A}}) \wedge (\beta_{\tilde{D}} > \beta_{\tilde{C}})] \Rightarrow \beta_{\tilde{B}} + \beta_{\tilde{D}} > \beta_{\tilde{A}} + \beta_{\tilde{C}}$;
- $[(\beta_{\tilde{B}} > \beta_{\tilde{A}}) \wedge (\beta_{\tilde{D}} = \beta_{\tilde{C}})] \Rightarrow \beta_{\tilde{B}} + \beta_{\tilde{D}} > \beta_{\tilde{A}} + \beta_{\tilde{C}}$;
- $[(\beta_{\tilde{B}} = \beta_{\tilde{A}}) \wedge (\beta_{\tilde{D}} > \beta_{\tilde{C}})] \Rightarrow \beta_{\tilde{B}} + \beta_{\tilde{D}} > \beta_{\tilde{A}} + \beta_{\tilde{C}}$;

i.e. $f(\tilde{A}+\tilde{C},\tilde{B}+\tilde{D}) \ge 0$;

- And since $[(\beta_{\tilde{B}} = \beta_{\tilde{A}}) \wedge (\beta_{\tilde{D}} = \beta_{\tilde{C}})]$ we have also $f(\tilde{A}+\tilde{C},\tilde{B}+\tilde{D}) \ge 0$.

If $f(\tilde{A},\tilde{B}) \geq 0$ and $k \geq 0$ and then by (3.2) and (3.3):
$f(k\tilde{A},k\tilde{B}) = \text{sign}(k) \times f(\tilde{A},\tilde{B}) \geq 0$ ■.

Theorem 3.2: If *f* is the DSCRF on an arbitrary linear set *X* then can construct a strict comparative relation $\Im$ on *X* correlative.
Proof: We definition a relation $\Im$ on *X*:
$x \underset{\Im}{\leq} y \Leftrightarrow f(x,y) \geq 0$
According to the definition 3.6, it is easy to prove $\Im$ is a comparative relation■.

4. Fuzzy Single-Source Shortest Path Problem

4.1 Design data model

Representation triangular fuzzy numbers:
To represent a set of triangular fuzzy numbers, we use class *TFN* with properties *core*, *left* and *right*.

```
class TFN {
  float core;
  float left;
  float right;
}
```

Representation graph:
Given a directed and connected graph $GF = \{V, E\}$ where $V = \{1,..,n\}$ is a set of n vertices of the graph and E is a set of edges. The weights of the edges are stored in an array $wf[\,]\,[\,]$ of type *TFN* where $wf[i, j]$ is the weight of the edges which connected between vertex *i* and vertex *j* of the graph.

4.2 Algorithm

Input: - Graph $GF = \{V, E\}$;
- Weighted array $wf[\,][\,]$;
- Source vertex $s \in V$;
- *f* is the DSCRF.

Output: The length of the shortest paths from source vertex $s \in V$ to all other vertices of the graph.
In this algorithm, we are using two functions:
Function $FAdd(\tilde{A},\tilde{B})$ where $\tilde{A},\tilde{B} \in TFN$ returns the sum of $\tilde{A}$ and $\tilde{B}$.
Function $IsLess(\tilde{A},\tilde{B})$ where $\tilde{A},\tilde{B} \in TFN$ returns *True* if $f(\tilde{A},\tilde{B}) > 0$ and returns *False* otherwise.
Like Dijkstra's algorithm, we use two arrays $df[\,] \in TFN$ and $u[\,]$ to hold label and status of a vertex which is reviewed or not reviewed.

At the beginning, we assign $df[s] = (0;0,0)$ and $df[v] = (\infty;0,0), u[v] = false\ \forall v \in V - s$.
The algorithm is shown in Figure 2:

```
for (int i=1; i<=n; i++) {

  // finding minimum label among unvisited vertices
  int v = -1;
  for (int j=1; j<=n; j++) {
    if (u[j] == false && (v == -1 || IsLess(df[j],df[v]))
    {
      v = j;
    }
  }

  // set as visited
  u[v]=true;

  for (int j=1; j<=n; j++) {
    if IsLess(0,wf[v][j]) {
      if IsLess(FAdd(df[v],wf[v][j]),df [j])=True {
        df[j] = FAdd(df[v],wf[v][j]);
      }
    }
  }
}
```

Fig. 2 The algorithm to finding fuzzy shortest path in a graph

4.3 Prove algorithm

Denote *U* is the set of vertices not yet reviewed.
The algorithm is proved by the mathematical induction. At step *k* we have the inductive hypothesis is:
(i) The label of $v \notin U$ is the length of the shortest path from vertex *s* to vertex *v* which is found at step *k*;
(ii) The label of $v \in U$ is the length of the shortest path from vertex *s* to vertex *v* and this path includes only *the vertices (isn't itself) not belong to U.*
- When *k*=1, *U*=*V*-*s*, the length of the shortest path from vertex *s* to all other vertices is infinite and the shortest path length from vertex s to itself is 0. That is mean basic step is true.
- Suppose that the inductive hypothesis is true at step *k*.
Call *v* is the vertex, review at *k*+1-th step. Then we have *df*[*v*] is minimum of the labels in the rest of *U*. From the inductive hypothesis implies when finish step *k* then the labels of vertices not belong to *U* is the minimum length of the shortest path from vertex *s* to it. The label of *v* is also the length of the shortest path from *s* to *v*. If this is not true then after *k*-th iteration, this is a path includes the vertices in *U* have a length be less than *df*[*v*]. Call *t* is the first vertex of the path and *t* is in *U*. Such, the length of the path from vertex *s* to vertex *t* is less than *df[v]* and include also the vertices not in *U*. That is a conflict with selection *v*. Therefore, at *k+1*-th iteration, we can confirm *(i)* is true. Call *r* is an arbitrary vertex in *U*, after *k+1*-th iteration. The shortest path from vertex *s* to vertex *r* includes only vertices not in *U*, and this path includes *v* or not. If it does not include *v* then as the inductive hypothesis the length is

df[v]. If *v* belongs to then the path is the path from *s* to *v* and adds edge *(v, r)*. As the properties of a strict comparative relation, the minimum weighted of this path is sum *df*[*v*] and *wf*[*v*,*r*]. It means that the label of vertex *r* is *df*[*v*]+*wf*[*v*,*r*]. This confirms (*ii*) is true.

4.4 Illustrative Example

Consider a real problem: Company X needs to ship goods from a source point to the 6 other points. The time for moving from A to B is defined by three statuses: Best (MO), Normal (ML) and Worst (MP). These values are given in Table 1:

Table 1: Estimated time between the points

Stretch of road	MO	ML	MP
1-2	2	6	12
1-4	5	12	20
2-3	7	16	26
2-5	19	30	39
3-5	5	14	19
4-3	2	10	17
4-6	25	33	41
5-6	9	9	10
5-7	15	21	31
6-7	5	12	15

The above problem is converted to the FSPP with the weights of the edges is the triangular fuzzy numbers.

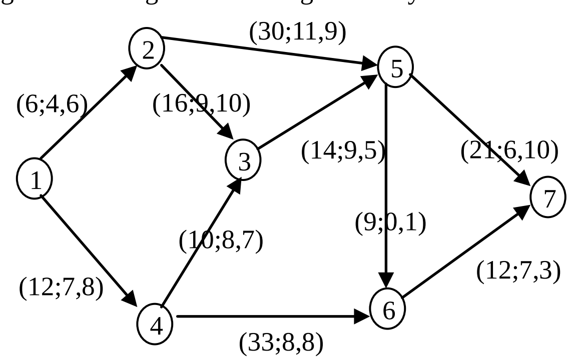

Fig. 3 Representation of weights of the edges

Where we use the DSCRF is:

$$f(\tilde{A},\tilde{B}) = \omega \operatorname{sign}(b-a) + \omega_1 \operatorname{sign}(\beta_{\tilde{B}} - \beta_{\tilde{A}})$$
$$+ \omega_2 \operatorname{sign}(\alpha_{\tilde{B}} - \alpha_{\tilde{A}})$$

with ω=0.8; ω_1=0.5 and ω_2=0.2.
The result of the iterations is represented in Table 2.

Table 2: Show the results of the algorithm

Step	Vertices have reviewed	Vertices have assigned
1	1	df[1]=(0;0,0) df[2]=(6;4,6) df[4]=(12;7,8)
2	1,2	df[1]=(0;0,0) df[2]=(6;4,6) df[4]=(12;7,8) df[3]=(22;13,16) df[5]=(36,15,15)
3	1,2,4	df[1]=(0;0,0) df[2]=(6;4,6) df[4]=(12;7,8) df[3]=(22;15,15) df[5]=(36;15,15) df[6]=(45;15,16)
4	1,2,4,3	df[1]=(0;0,0) df[2]=(6;4,6) df[4]=(12;7,8) df[3]=(22;15,15) df[5]=(36;15,15) df[6]=(45;15,16)
5	1,2,4,3,5	df[1]=(0;0,0) df[2]=(6;4,6) df[4]=(12;7,8) df[3]=(22;15,15) df[5]=(36;15,15) df[6]=(45;15,16) df[7]=(57;21,25)
6	1,2,4,3,5,6	df[1]=(0;0,0) df[2]=(6;4,6) df[4]=(12;7,8) df[3]=(22;15,15) df[5]=(36;15,15) df[6]=(45;15,16) df[7]=(57;22,19)
7	1,2,4,3,5,6,7	df[1]=(0;0,0) df[2]=(6;4,6) df[4]=(12;7,8) df[3]=(22;15,15) df[5]=(36;15,15) df[6]=(45;15,16) df[7]=(57;22,19)

The lengths of the shortest paths from vertex 1 to all other vertices are shown in Table 3.

Table 3: The Fuzzy Shortest Paths from vertex 1 to all other vertices

Destination	Length of path	Road
1	(0;0,0)	1-1
2	(6;4,6)	1-2
3	(22;15,15)	1-4-3
4	(12;7,8)	1-4
5	(36;15,15)	1-2-5
6	(45;15,16)	1-2-5-6
7	(57;22,19)	1-2-5-6-7

5. Conclusion

This paper focuses on solving the single-source shortest path problem in the case of the weights of the edges are non-negative and are represented by triangular fuzzy numbers. Here we use new concept is called the Defined Strict Comparative Relation Function (DSCRF) on the set of triangular fuzzy numbers and apply expanded Dijkstra's algorithm for this case.

References

[1] D. Dubois and H. Prade, Fuzzy Sets and Systems: Theory and Applications, New York: Academic Press, 1980.
[2] S.Okada and T. Soper, “A shortest path problem on a network with fuzzy arc lengths,” Fuzzy Sets and Systems, 109, 2000, pp. 129-140.
[3] Tzung-Nan Chuang and Jung-Yuan Kung, "A new algorithm for the discrete fuzzy shortest path problem in a network", Applied Mathematics and Computation, 174, 2006, pp. 660-668.
[4] L. Sujatha and S. Elizabeth, "Fuzzy Shortest Path Problem Based on Similarly Degree", Applied Mathematical Sciences, Vol. 5, No. 66, 2011, pp.3263-3276.
[5] Amit Kumar and Manjot Kaur, "A New Algorithm for Solving Shortest Path Problem on a Network with Imprecise Edge Weight", Applications and Applied Mathematics: An International Journal, Vol.6, Issue 2, 2011, pp. 602-619.
[6] N.T.Luan and V.T.T Huyen, "The Comparative Relation and Its Application in solving Fuzzy Linear Programming Problem", International Journal of Computer Science Issues, Vol. 9, Issue 4, No. 2, 2012, pp. 167-174.
[7] Kairanbay Magzhan and Hajar Mat Jani, "A Review and Evaluations of Shortest Path Algorithms", International Journal of Scientific and Technology Research, Vol. 2, Issue 6, 2013, pp. 99-104.

Vu Thi Thu Huyen is a lecturer in the Department of Information Technology at the University of Economic and Technical Industries, Vietnam. She received the Master of IT in 2009 at Le Qui Don Technical University, Vietnam where she is now a student of doctor course. Her area of research is fuzzy optimization.

Nguyen Thien Luan is a lecturer of the Thang Long University, Hanoi, Vietnam. His research interests include fuzzy logic, fuzzy optimization, image processing, communications and network security. He has authored or co-authored of more than 30 scientific articles, book chapters and chaired many scientific research projects, in areas of his research. He received his Ph.D. (1989) and Associate Professor (2003).

Le Minh Tuan is a lecturer of Thang Long University, Hanoi, Vietnam. He received Bachelor degree in Mathematics and Informatics at Thang Long University and Master degree in Computer Science at Le Qui Don Technical University, Vietnam in 2004 and 2008 respectively. His research related to artificial intelligent, data mining, and fuzzy logic.