Particle swarm optimisation algorithm for solving shortest path problems with mixed fuzzy arc weights

Ali Ebrahimnejad*

Department of Mathematics,
Qaemshahr Branch,
Islamic Azad University,
P.O. Box 163, Qaemshahr, Iran
Fax: +98-1232240091
Email: aemarzoun@gmail.com
Email: a.ebrahimnejad@qaemshahriau.ac.ir
*Corresponding author

Zahra Karimnejad and Hamidreza Alrezaamiri

Department of Computer Sciences,
Islamic Azad University,
Babol Branch, Baol, Iran
Fax: +98-1132415007
Email: kz69@rocketmail.com
Email: hamidreza.alreza@baboliau.ac.ir

Abstract: Shortest path problem is one of the most fundamental components in the fields of transportation and communication networks. This paper concentrates on a shortest path problem on a network where arc weights are represented by different kinds of fuzzy numbers. Recently, a genetic algorithm has been proposed for finding the shortest path in a network with mixed fuzzy arc weights due to the complexity of the addition of various fuzzy numbers for larger problems. In this paper, a particle swarm optimisation (PSO) algorithm in fuzzy environment is used for the same due to its superior convergence speed. The main contribution of this paper is the reduction of the time complexity of the existing genetic algorithm. Additionally, to compare the obtained results of the proposed PSO algorithm with those of the existing algorithm, two shortest path problems having mixed fuzzy arc weights are solved. The comparative examples illustrate that the algorithm proposed in this paper is more efficient than the existing algorithm in terms of time complexity.

Keywords: shortest path problem; fuzzy numbers; particle swarm optimisation algorithm.

Reference to this paper should be made as follows: Ebrahimnejad, A., Karimnejad, Z. and Alrezaamiri, H. (2015) 'Particle swarm optimisation algorithm for solving shortest path problems with mixed fuzzy arc weights', *Int. J. Applied Decision Sciences*, Vol. 8, No. 2, pp.203–222.

Biographical notes: Ali Ebrahimnejad serves as an Assistant Professor at Mathematics Department, Qaemshahr Branch, Islamic Azad University. He also serves as a reviewer for several journals such as *Fuzzy Sets and Systems*, *Journal of Intelligent and Fuzzy Systems*, *Int. J. Fuzzy Systems*, *Computers and*

Industrial Engineering and *European Journal of Operational Research*. He is on the editorial board of *International Journal of Fuzzy System Applications*, *International Journal of Enterprise Information Systems*, *International Journal of Strategic Decision Sciences* and *International Journal of Information and Decision Sciences*. His research interests include operations research, network flows, data envelopment analysis and fuzzy linear programming.

Zahra Karimnejad is an MSc student in Computer Software Engineering at Islamic Azad University, Babol Branch, under supervisory of Dr. Ali Ebrahimnejad. She received her BS from Mazandran University of Science and Technology, Babol, Iran. Her major research interests are fuzzy set theory, heuristic algorithms, RFID and WSN.

Hamidreza Alrezaamiri obtained his MSc in Computer Software Engineering at the Islamic Azad University, Babol Branch, Iran and his BS degree at Iran University of Science and Technology. His major research interests are fuzzy set theory, heuristic algorithms, RFID and WSN.

1 Introduction

Shortest path (SP) problem is an important network structured linear programming problem that arises in several contexts and has deservedly received a great deal of attention in the literature (Aboutahoun, 2010; Rudniy et al., 2010; Chitra et al., 2011; Nikulin and Iftikhar, 2012; Dong et al., 2013; Olya 2014a, 2014b; Trevizan and Veloso, 2014; Lissovoi and Witt, 2015; Lai, 2015; Ardakani and Tavana, 2015; Grigoryan and Harutyunyan, 2015; Rezvanian and Meybodi, 2015; Aridhi et al., 2015). The central concept in the problem is to find a path with minimum weight (cost, time or length) between two specified nodes. This problem can be used for a wide variety of situations such as transportation, routing, communications, supply chain management and models involving agents. In addition SP problems arise frequently as sub-problems when solving many combinatorial and network optimisation problems. Even though SP problems are relatively easy to solve, the design and analysis of most algorithms for solving them require considerable ingenuity (Ahuja, 1993).

In general, SP problems are solved with the assumptions that the weights of arcs are specified in a precise way, i.e., in crisp environment. However, in many cases the decision maker has no crisp information about the coefficients belonging to the SP problem. In these cases, using fuzzy numbers for formulation of the problem is quite appropriate and fuzzy SP problem appears in a natural way.

The overall contribution of this study is summarised as follows:

1 The particle swarm optimisation (PSO) algorithm proposed in this study gives both fuzzy SP and the corresponding SP fuzzy weight in the fuzzy network under consideration.

2 The proposed algorithm, similar to the competing algorithms in the literature, while being practically simple, has the flexibility to consider a mixture of various types of fuzzy arc lengths in a general network.

3 The time complexity of the proposed algorithm is reduced very much compared to some existing algorithms commonly used in the literature.

This paper is structured as follows. In Section 2, basic concepts and definitions of fuzzy set theory, computing α-cut for fuzzy numbers and the distance between fuzzy numbers are reviewed. A PSO algorithm for the fuzzy SP in a network having mixed fuzzy arc weights is presented in Section 3. Two comparative examples are illustrated in Section 4. Finally, conclusions and discussions are given in Section 5.

2 Literature review

Since the SP problem can be formulated as a linear programming where the constraints have a special structure, one straightforward method for solving fuzzy SP problems is utilising the fuzzy linear programming solution techniques(Ebrahimnejad 2011a, 2011b, 2013a, 2013b, 2014; Ebrahimnejad et al., 2013; Khalili-Damghani et al., 2013; Sahebjamnia et al., 2013; Kazemi et al., 2014; Messaoudi and Rebaï, 2014; Ebrahimnejad and Tavana, 2014; Ebrahimnejad and Verdegay, 2014a, 2014b). However, because of its very special structure, a host of specialised algorithms have been proposed for the fuzzy SP problems in the literature.

An overview of the articles which studied fuzzy SP problem and discussed related methodologies can be summarised as follows. Dubois and Prade (1980) first analysed the fuzzy SP (FFP) problem and considered extensions of the classic Floyd and Ford-Moore-Bellman (FMB) algorithms that return distances without an associated path. Klein (1991) introduced new models based on fuzzy SPs and also proposed a general algorithm based on dynamic programming to solve the new models. In addition the SFP algorithms were analysed in terms of sub-modular functions in that paper. Lin and Chern (1993) considered the case that the arc weights are fuzzy numbers and proposed an algorithm for finding the single most vital arc in a network as being that whose removal from the path results in an increase in cost. The neural networks were introduced for solving SP problems by Li et al. (1996). Okada and Soper (2000) concentrated on a SP problem in a network with fuzzy arc weights. Then they proposed an algorithm to obtain all Pareto Optimal paths from the specified node to every other node by introducing a concept of Pareto Optimal path based on an order relation between fuzzy numbers. Following the idea of finding a fuzzy set solution, Okada (2004) presented an algorithm to determine the degree of possibility for each arc on the SP. Chuang and Kung (2005) proposed a heuristic procedure to find the FSP length among all possible paths in a network. Chuang and Kung (2006) proposed a new algorithm that gives the FSP length and the corresponding SP in a discrete FSP problem. Hernandes et al. (2007) considered a generic algorithm for solving FSP problem that can be implemented using any fuzzy numbers ranking index chosen by the decision-maker. Ji et al. (2007) introduced three types of models for FSP problem based on the concepts of expected SP, α-SP and the most SP in fuzzy environment. They also proposed a hybrid intelligent algorithm integrating simulation and genetic algorithm in order to solve these models. Gao (2011) proved that there exists an equivalence relation between the α-SP of an uncertain network and the SP of the corresponding deterministic network. Mahdavi et al. (2009) focused on finding shortest chains in a graph with fuzzy distance for every arc and proposed a dynamic programming approach to solve the fuzzy shortest chain problem using a suitable ranking method. Kumar and Kaur (2011) presented a new algorithm for solving SP problem on a network with imprecise arc weights. Dou et al. (2012) applied an approach to select the SP in multi-constrained network using multi-criteria decision method based on vague similarity measure. Deng et al. (2012) extended the Dijkstra algorithm to solve the SP problem with fuzzy arc weights. Their proposed method to find the SP under fuzzy arc lengths is based on the graded mean integration representation of fuzzy numbers. Zhang et al. (2013) proposed a biologically inspired algorithm called Fuzzy *Physarum* Algorithm for fuzzy SP problems based on path finding model. Rangasamy et al. (2013) proposed a method for finding the shortest hyper path in an intuitionistic fuzzy weighted hypergraph using the scores and accuracy of intuitionistic fuzzy numbers. Tajdin et al. (2010) designed an algorithm for computing a SP in a network having various types of fuzzy arc lengths. They used an α-cut approach to compute the addition of various fuzzy numbers as arc weights. After that, Hassanzadeh et al. (2013) presented a genetic algorithm for finding the SP in the network due to the complexity of the addition of various fuzzy numbers for larger problems. In this paper, we use a population-based metaheuristic optimisation algorithm, namely PSO, to approximate a short path on the same network, where arcs are weighted with different kinds of fuzzy numbers.

3 Preliminaries

In this section, some basic definitions and arithmetic operations on fuzzy numbers are presented (Dubois and Prade, 1980; Tajdin et al., 2010; Hassanzadeh et al., 2013).

3.1 Fuzzy numbers

Definition 1: Let X be the universal set. The set $\tilde{a}$ is called a fuzzy set in X if $\tilde{a}$ is a set of ordered pairs $\tilde{a} = \{(x, \mu_{\tilde{a}}(x)) \mid x \in X\}$, where $\mu_{\tilde{a}}(.)$ is a membership function of $\tilde{a}$ and assigns to each element $x \in X$ a real number $\mu_{\tilde{a}}(x)$ in the interval [0,1].

Definition 2: Given a fuzzy set $\tilde{a}$ defined on X and any number $\alpha \in [0,1]$, the α-cut is the crisp set $[\tilde{a}]_\alpha = \{x \in X; \mu_{\tilde{a}}(x) \geq \alpha\} = [\tilde{a}_\alpha^L, \tilde{a}_\alpha^R]$.

Definition 3: A fuzzy number is a convex normalised fuzzy set of the real line $\mathbb{R}$, whose membership function is piecewise continuous.

Definition 4: A function L: $[0, \infty) \to [0,1]$ (or R: $[0, \infty) \to [0,1]$) is said to be reference function of fuzzy numbers if and only if

1 $L(0) = 1$ ($R(0) = 1$)

2 L(or R) is non-increasing on $[0, \infty)$.

Example 1: The commonly used linear reference functions and nonlinear reference functions with parameter q, denoted as RF_q, are summarised as follows:

1 linear: $\max\{0, 1 - x\}$

2 power: $RF_q = \max\{0, 1 - x^q\}, q > 0$

3 exponential power: $RF_q = e^{-x^q}, q > 0$

4 rational: $RF_q = \dfrac{1}{1+x^q}, q > 0$

Definition 5: A fuzzy number $\tilde{a}$, denoted by $\tilde{a} = (m, a, b)_{LR}$, is called an *LR* fuzzy number if the membership function $\mu_{\tilde{a}}(x)$ is given by

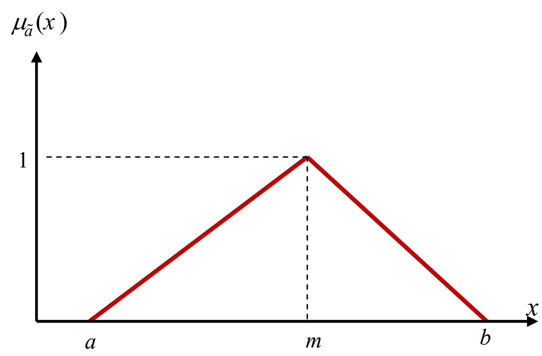
$$\mu_{\tilde{a}}(x) = \begin{cases} L\left(\dfrac{m-x}{a}\right), & x \leq m, \\ R\left(\dfrac{x-m}{b}\right), & x \geq m. \end{cases}$$

The set of *LR* fuzzy numbers on real line is denoted by $\ell\mathfrak{R}(\mathbb{R})$.

It should be noted that if $L(x) = R(x) = \max\{0, 1 - x\}$ then the *LR* fuzzy number $\tilde{a} = (m, a, b)_{LR}$ is denoted by $\tilde{a} = (m, a, b)$ and is called a triangular fuzzy number with the following membership function (see Figure 1):

$$\mu_{\tilde{a}}(x) = \begin{cases} \dfrac{x-(m-a)}{a}, & x \leq m, \\ \dfrac{(m+b)-x}{b}, & x \geq m. \end{cases}$$

Figure 1 Triangular fuzzy number $\tilde{a} = (m, a, b)$ (see online version for colours)

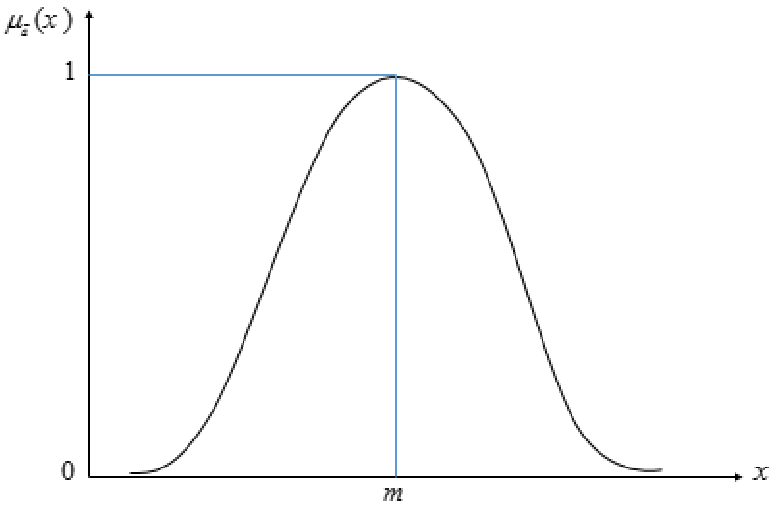


Definition 6: If $L(x) = R(x) = e^{-x^2}$ then the *LR* fuzzy number $\tilde{a} = (m, a, b)_{LR}$ is called a normal fuzzy number fuzzy number and shown by $\tilde{a} = (m, \sigma)$ with the following membership function (see Figure 2):

$$\mu_{\tilde{a}}(x) = e^{-\left(\frac{x-m}{\sigma}\right)^2}, \quad x \in \mathbb{R}$$

where *m* and *σ* are mean and standard deviation, respectively.

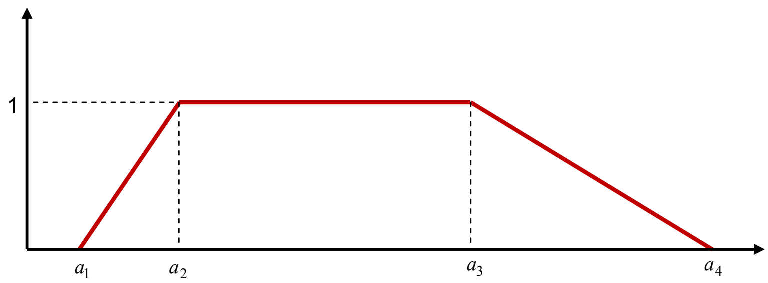
Figure 2 Normal fuzzy number $\tilde{a} = (m, \sigma)$ (see online version for colours)



Definition 7: A trapezoidal fuzzy number $\tilde{a}$ is denoted by $\tilde{a} = (a_1, a_2, a_3, a_4)$, with the following membership function (see Figure 3):

$$\mu_{\tilde{a}}(x) = \begin{cases} \dfrac{x - a_1}{a_2 - a_1}, & a_1 \le x \le a_2, \\ 1, & a_2 \le x \le a_3, \\ \dfrac{a_4 - x}{a_4 - a_3}, & a_3 \le x \le a_4. \end{cases}$$

Figure 3 A trapezoidal fuzzy number $\tilde{a} = (a_1, a_2, a_3, a_4)$ (see online version for colours)



3.2 *α-cut of fuzzy numbers*

One approach to develop the arithmetic of fuzzy numbers is based on α-cuts.

Definition 8: Let $\tilde{a} = (m, a, b)_{LR}$ be an LR fuzzy number with L and R reference functions. Then, the α-cut of $\tilde{a}$ is given by

$$[\tilde{a}]_\alpha = \left[\tilde{a}^L_\alpha, \tilde{a}^R_\alpha\right] = \left[m - aL^{-1}(\alpha), m + bR^{-1}(\alpha)\right].$$

Definition 9: Let $\tilde{a} = (m, \sigma)$ be a normal fuzzy number. The α-cut of $\tilde{a}$ is given by

$$[\tilde{a}]_\alpha = \left[\tilde{a}^L_\alpha, \tilde{a}^R_\alpha\right] = \left[m - \sigma\sqrt{-\ln(\alpha)}, m + \sigma\sqrt{-\ln(\alpha)}\right].$$

Definition 10: Let $\tilde{a} = (a_1, a_2, a_3, a_4)$ be a trapezoidal fuzzy number. The α-cut of $\tilde{a}$ is given by

$$[\tilde{a}]_\alpha = \left[\tilde{a}^L_\alpha, \tilde{a}^R_\alpha\right] = \left[(a_2 - a_1)\alpha + a_1, a_4 - (a_4 - a_3)\alpha\right].$$

3.3 Approximate summation of mixed fuzzy numbers

Hassanzadeh et al. (2013) proposed an exponential membership function for the approximating sum of trapezoidal and normal fuzzy numbers. For doing this, they divided the α-interval, [0, 1], into n subinterval by letting $\alpha_0 = 0$ and

$$\alpha_i = \alpha_{i-1} + \Delta\alpha_i, \ \Delta\alpha_i = \frac{1}{n}, \ i = 1, 2, \ldots, n.$$

When the normal fuzzy numbers are considered, it is not reasonable to set α equal to 0. In this case, we let $\alpha \in (0, 1]$.

Let $\tilde{a} = (a_1, a_2, a_3, a_4)$ and $\tilde{b} = (m, \sigma)$ be trapezoidal and normal fuzzy numbers, respectively. Given $\alpha_i \in (0,1]$, $1 \le i \le n$, the α_i-cut sum of these fuzzy numbers using Definitions 6 and 7 is obtained as follows:

$$\begin{aligned}[\tilde{c}]_{\alpha_i} &= [\tilde{a}]_{\alpha_i} + [\tilde{b}]_{\alpha_i} = \left[\tilde{a}^L_{\alpha_i} + \tilde{b}^L_{\alpha_i}, \tilde{a}^R_{\alpha_i} + \tilde{b}^R_{\alpha_i}\right] \Rightarrow \left[\tilde{c}^L_{\alpha_i}, \tilde{c}^R_{\alpha_i}\right] \\ &= \left[(a_2 - a_1)\alpha_i + a_1 + m - \sigma\sqrt{-\ln(\alpha_i)}, a_4 - (a_4 - a_3)\alpha_i + m + \sigma\sqrt{-\ln(\alpha_i)}\right]\end{aligned} \tag{1}$$

Corresponding to equation (1), using the α_i, $1 \le i \le n$, n points for $\tilde{c}^L_{\alpha_i}$ and n points for $\tilde{c}^R_{\alpha_i}$ are gained.

Hassanzadeh et al. (2013) approximated the membership function of the sum using the resulting points via α-cut and Crammer's approach for fitting a membership function for the sum. Let $x_i = \tilde{c}^R_{\alpha_i}$ and $y_i = \mu(\tilde{c}^R_{\alpha_i})$, and for n points (x_i, y_i), consider the fitting model to be as $y = e^{-\left(\frac{x-\lambda}{\beta}\right)^2}$. They proposed a least squares model to approximate the right membership function for the considered addition, and determined the unknown parameters λ and β as follows (Tajdin et al., 2010; Hassanzadeh et al., 2013):

$$\beta = \frac{n\sum_i \left(x_i \times \sqrt{-\ln y_i}\right) - \sum_i \sqrt{-\ln y_i} \times \sum_i x_i}{-n\sum_i \sqrt{-\ln y_i} - \sum_i \sqrt{-\ln y_i} \times \sum_i \sqrt{-\ln y_i}} \tag{2}$$

$$\lambda = \frac{\sum_i \ln y_i \left(-\sum_i x_i\right) - \sum_i \left(x_i \times \sqrt{-\ln y_i}\right) \times \sum_i \sqrt{-\ln y_i}}{-n\sum_i \sqrt{-\ln y_i} - \sum_i \sqrt{-\ln y_i} \times \sum_i \sqrt{-\ln y_i}} \tag{3}$$

In a similar way, let $x_i = \tilde{c}^L_{\alpha_i}$ and $y_i = \mu(\tilde{c}^L_{\alpha_i})$, and consider the fitting model $y = e^{-\left(\frac{x-\lambda'}{\beta'}\right)^2}$. The least squares model to approximate the left membership function for the considered addition gives the unknown parameters λ' and β' as follows (Tajdin et al., 2010; Hassanzadeh et al., 2013):

$$\beta' = \frac{n\sum_i \left(x_i \times \sqrt{-\ln y_i}\right) - \sum_i \sqrt{-\ln y_i} \times \sum_i x_i}{n\sum_i \sqrt{\ln y_i} + \sum_i \sqrt{-\ln y_i} \times \sum_i \sqrt{-\ln y_i}} \tag{4}$$

$$\lambda' = \frac{\sum_i \ln y_i \times \sum_i x_i + \sum_i \left(x_i \times \sqrt{-\ln y_i}\right) \times \sum_i \sqrt{-\ln y_i}}{n\sum_i \sqrt{\ln y_i} + \sum_i \sqrt{-\ln y_i} \times \sum_i \sqrt{-\ln y_i}} \tag{5}$$

Therefore, the approximate membership function for the approximating sum of trapezoidal and normal fuzzy numbers is computed by:

$$\mu_{\tilde{c}}(x) = \begin{cases} e^{-\left(\frac{\lambda'-x}{\beta'}\right)^2}, & x < \lambda', \\ 1, & \lambda' \le x \le \lambda, \\ e^{-\left(\frac{x-\lambda}{\beta}\right)^2}, & x > \lambda. \end{cases} \tag{6}$$

3.4 Distance between mixed fuzzy numbers

In this subsection, the distance between two fuzzy numbers using the resulting points from the α-cut is reviewed (Tajdin et al., 2010; Hassanzadeh et al., 2013).

Given two fuzzy numbers $\tilde{a}$ and $\tilde{b}$, the $D_{p,q}$-distance between them is defined as follows:

$$D_{p,q}\left(\tilde{a},\tilde{b}\right) = \begin{cases} \left[(1-q)\int_0^1 \left|a_\alpha^- - b_\alpha^-\right|^p d\alpha + q\int_0^1 \left|a_\alpha^+ - b_\alpha^+\right|^p d\alpha \right], & p < \infty \\ (1-q) \sup\limits_{0<\alpha\le 1} \left|a_\alpha^- - b_\alpha^-\right| + q \inf\limits_{0<\alpha\le 1} \left|a_\alpha^+ - b_\alpha^+\right|, & p = \infty \end{cases} \tag{7}$$

where the first parameter p denotes the priority weight attributed to the end points of the support; for instance, the a_α^- and a_α^+ of the fuzzy numbers. If the expert has no preference, $D_{p,\frac{1}{2}}$ is used. The second parameter q determines the analytical properties of $D_{p,q}$. For two fuzzy numbers $\tilde{a}$ and $\tilde{b}$, the $D_{p,q}$ is proportional to

$$D_{p,q}\left(\tilde{a},\tilde{b}\right)=\left[(1-q)\sum_{i=1}^{n}\left|a_{\alpha_i}^{-}-b_{\alpha_i}^{-}\right|^{p}+q\sum_{i=1}^{n}\left|a_{\alpha_i}^{+}-b_{\alpha_i}^{+}\right|^{p}\right]^{\frac{1}{p}} \tag{8}$$

If $q=\frac{1}{2}$ and $p = 2$, then the above equation turns into:

$$D_{2,\frac{1}{2}}\left(\tilde{a},\tilde{b}\right)=\sqrt{\left[\frac{1}{2}\sum_{i=1}^{n}\left|a_{\alpha_i}^{-}-b_{\alpha_i}^{-}\right|^{2}+\frac{1}{2}\sum_{i=1}^{n}\left|a_{\alpha_i}^{+}-b_{\alpha_i}^{+}\right|^{2}\right]} \tag{9}$$

To compare two fuzzy arc weights $\tilde{a}$ and $\tilde{b}$ using the α_i-cuts for their approximations, since they are supposed to represent positive values, they are compared with $\tilde{0}=(0,0,\ldots,0)$. In fact, formula (9) is used to compute $D_{2,\frac{1}{2}}(\tilde{a},\tilde{0})$ and $D_{2,\frac{1}{2}}(\tilde{b},\tilde{0})$. In this case, we say that $\tilde{a}\,\tilde{\leq}\,\tilde{b}$ if and if $D_{2,\frac{1}{2}}(\tilde{a},\tilde{0})\leq D_{2,\frac{1}{2}}(\tilde{b},\tilde{0})$.

4 PSO algorithm for solving fuzzy SP

In this section, a PSO algorithm is presented for finding the SP and the corresponding SP weight in the fuzzy network under consideration. PSO algorithm is a simple algorithm with only a few parameters to be adjusted during the optimisation process, rendering it compatible with any modern computer language. Also, it is more efficient than other evolutionary algorithms due to its superior convergence speed (Yang and Zhang, 2010).

4.1 Particle swarm optimisation

PSO algorithm has been presented for the first time in 1995 by Kennedy and Eberhart (1995) as an evolutionary computation technique. In recent years this heuristic algorithm have been proposed to solve the real life problems related to many engineering applications, which have achieved better results in terms of computational and time complexity (Gao et al., 2013; Zhang et al., 2014; Tang et al., 2014; Mahi et al., 2015; Ardizzon et al., 2015; Zhang et al., 2015; Sadeghzadeh et al., 2015; Moraes et al., 2015).

The PSO algorithm is inspired from the swarm movement of the birds searching for food. It is a population-based algorithm in which its individuals (known as particles) encode potential solutions to n-dimensional optimisation problems and explore the search space through cooperation with other particles. The cooperation takes place by communicating the best solutions found so far and moving towards them.

Particles have a position vector $x(t)$ that encodes a potential solution to the problem, and a velocity vector $v(t)$ that determines the change in the position according to

$$x(t+1)=v(t+1)+x(t) \tag{10}$$

The velocity vector balances the trade-off between exploration and exploitation of the search: high velocities result in large changes in the position of the particles (exploration), whereas low velocities produce small changes (exploitation). The velocity vector for each iteration is computed as follows (Calazan et al., 2014):

$$v(t+1) = wv(t) + c_1 r_1 [x(t) - pbest] + c_2 r_2 [x(t) - gbest] \quad (11)$$

where w is the inertia of the particle (Shi and Eberhart 1998), c_1 and c_2 are positive acceleration coefficients (cognitive and social components) that weigh the importance of the personal and neighbourhood knowledge, r_1 and r_2 are random values in [0,1], *pbest* is the best position (the position giving the best fitness value) found so far by the particle itself, and *gbest* is the best one found by its neighbourhood.

It should be noted that w or inertia factor, is a parameter that controls impact of current velocity on the new velocity and plays the role of balancing the global search and local search. This factor can be computed according to following formula:

$$w = w_{\max} - \frac{w_{\max} - w_{\min}}{iter_{\max}} \times iter \quad (12)$$

where $w_{\max}$, $w_{\min}$, $iter_{\max}$ and $iter$ are initial weight, final weight, maximum iteration number and current iteration number, respectively.

4.2 Finding the SP by PSO

It should be noted that the main elements of PSO algorithm are constituted of two main components:

- population initialisation
- updating the velocity and position equations.

In what follows, we describe how to apply these elements for obtaining fuzzy SP.

4.2.1 Population initialisation

For finding the SP through PSO algorithm, at first the initial swarm should be established. The initial swarm is consisted of some particles that each particle is indeed a path from the origin of network to the destination of network. After establishing of the initial swarm that specifies the position of particles, the initial velocity of each particle should be established too. The velocity of each particle is also a path. For establishing the path in a network, the vicinity matrix to that network is required. Corresponding to each network, first the vicinity matrix is established and then by Algorithm 1 the velocity and the position of each article are established.

Algorithm 1 Producing of primary population

1	Determine the vicinity matrix of directed network $G = (V, E)$, give $particle - size$ and set $q = 1$.
2	Set $i = 1$, $m = 1$ and $p(m) = 1$.
3	Define $a^1(i) = \{j \mid (i, j) \in A, a_{ij} = 1\}$ and select a member of it, say j. Let $m = m + 1$ and $p(m) = j$.
4	If $j \neq n$ then let $i = j$ and go to (3).
5	Save the produced path using the labels in the labelling vector p. Let $q = q + 1$.
6	If $q \leq particle - size$ then go to (2) else stop.

4.2.2 Updating the velocity and position equations

It should be noted that PSO is a kind of evolutionary algorithm to find optimal solutions for continuous optimisation problems. But, SP problem with different fuzzy arc weights is a discrete optimisation problem. Thus, standard PSO equations are not able to generate discrete values since positions are real-valued and it is required to use heuristic method to overcome this shortcoming in solving fuzzy SP problem. For overcoming the problem of updating the velocity and position equations, crossover operator is added to the proposed discrete PSO algorithm for solving fuzzy SP problem. The applied crossover operator acts exactly like that one used in genetic algorithm (Hassanzadeh et al., 2013) and combines the information corresponding to two parents (articles) to generate two children. In what follows, the generation process of children is explored.

In the standard PSO algorithm, velocity equation is based on equation (10) in which three factors c_1r_1, c_2r_2 and w represent the impact amount of *pbest*, *gbest* and $v(t)$, respectively. In our proposed heuristic method, these factors are used to choose parents in crossover operator. In each updating step of the velocity equation, among the three parameters of c_1r_1, c_2r_2 and w, two factors having the most values, are selected as parents and the corresponding vectors execute the crossover operator. In this case, the result of crossover is a new path that is saved as $v(t + 1)$. In addition, in each updating step of the position equation, another crossover operator is executed between the current position of particle $x(t)$ and new velocity (child generated from the previous crossover) of particle $v(t + 1)$. For example, if $c_1r_1 = 0.9$, $c_2r_2 = 0.15$ and $w = 0.62$, since $c_1r_1 = 0.9$ and $w = 0.62$ have the higher values among these three values, then the *pbest* and w vectors execute crossover operator and their child is selected as $v(t + 1)$. Also, for updating the position equation, the vectors $v(t + 1)$ and $x(t)$ execute the crossover operator in order to obtain new position (path) $x(t + 1)$.

4.4.2.1 Crossover operator

The crossover operator in the genetic and PSO algorithms has the same process. This means that the standard used crossovers in genetic algorithm such as one-point, two-point, and uniform can be used in PSO models. Two paths, called parents, are randomly selected from the population.

The number of paths for the crossover operator is equal to the product of population size and crossover operator's rate. Then, one or two common members are selected and different sections of the codes for parents are modified. Hence, two new children (new path) are generated. It is obvious that the generated paths are feasible. If at least one member is common, then one-point crossover is performed and if at least two members are common, then two-point crossover is performed. In addition, if there is no common member among parents, a new path is obtained by Algorithm 1 and is considered as the result of crossover.

4.2.2.2 Fitness calculating

This value is determined by aggregating the arcs included in the path, where to sum the various arcs, equation (3) is applied. The result of the addition is a set of α-cut points. Then, for comparison of path values, the distance function $D_{2,\frac{1}{2}}$ is used as explained before. The values of $D_{2,\frac{1}{2}}$ is the path length and the minimum possible value in the network is the SP length (Hassanzadeh et al., 2013).

5 Comparative examples

In this section, to show the advantages of the proposed PSO algorithm over the existing genetic algorithm (Hassanzadeh et al., 2013), the numerical examples given in Hassanzadeh et al. (2013) are solved by using the proposed algorithm and the time complexity results of existing and the proposed algorithms are compared.

Example 5.1: Let us consider the network in Fig 4 with mixed fuzzy arc weights as given in Table 1. There are 11 nodes and 25 arcs in the network. Number of particles in the PSO algorithm and number of chromosomes in the genetic algorithm (Hassanzadeh et al., 2013) are 10. Number of iterations in both of these algorithms is same.

Figure 4 The network for Example 1

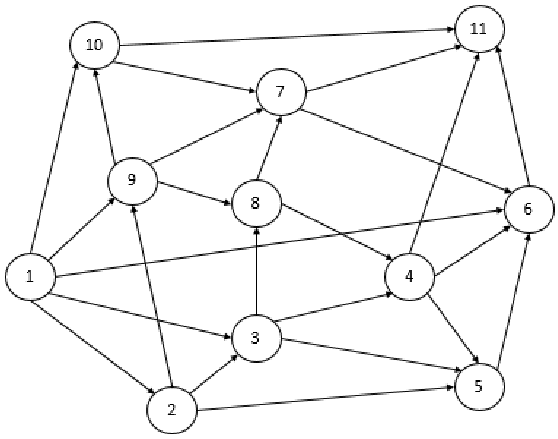


Table 1 The arc weights for Example 1

Arc	*Fuzzy number*	*Arc*	*Fuzzy number*	*Arc*	*Fuzzy number*
(1, 2)	(800, 820, 840)	(3, 5)	(730, 748, 870)	(8, 4)	(710, 730, 835)
(1, 3)	(35, 11)	(3, 8)	(42, 14)	(8, 7)	(230, 242, 355)
(1, 6)	(650, 677, 783)	(4, 5)	(190, 199, 310)	(9, 7)	(120, 130, 250)
(1, 9)	(290, 300, 350)	(4, 6)	(310, 340, 460)	(9, 8)	(13, 4)
(1, 10)	(420, 450, 570)	(4, 11)	(71, 23)	(9, 10)	(23, 7)
(2, 3)	(180, 186, 293)	(5, 6)	(610, 660, 790)	(10, 7)	(330, 342, 450)
(2, 5)	(495, 510, 625)	(6, 11)	(23, 7)	(10, 11)	(125, 41)
(2, 9)	(90, 30)	(6, 7)	(390, 410, 540)	(3, 4)	(650, 667, 983)
(7, 11)	(45, 15)				

The corresponding fuzzy SP problem has been solved 10 times using the proposed PSO algorithm and the existing genetic algorithm (Hassanzadeh et al., 2013). The results are given in Table 2.

Table 2 Information corresponding to ten runs of Example 1

	Generation	*Shortest path*	*Number of iteration to converge*		*Convergence time span (s)*		*Total time (s)*	
			GA	*PSO*	*GA*	*PSO*	*GA*	*PSO*
1	80	1-3-8-7-11	6	1	0.61	0.08	6.84	3.70
2	80	1-3-8-7-11	7	1	0.58	0.11	6.92	4.00
3	80	1-3-8-7-11	2	3	0.20	0.18	6.94	3.57
4	80	1-3-8-7-11	2	5	0.31	0.25	7.23	3.68
5	80	1-3-8-7-11	14	6	1.43	0.29	7.02	3.77
6	120	1-3-8-7-11	6	2	0.54	0.30	10.61	5.98
7	120	1-3-8-7-11	12	1	0.91	0.19	10.26	5.50
8	120	1-3-8-7-11	1	9	0.13	0.53	10.65	5.65
9	120	1-3-8-7-11	7	7	0.51	0.41	10.16	5.53
10	120	1-3-8-7-11	11	4	0.88	0.38	10.55	5.88
Min	-	-	1	1	0.13	0.11	6.84	3.57
Max	-	-	14	6	1.43	0.53	10.65	5.98
Mean	-	-	6.8	3.1	0.61	0.272	8.718	4.726

The fuzzy SP found by the proposed PSO algorithms is 1 →3→8→7→ 11matching the results of (Hassanzadeh et al., 2013).However, using PSO algorithm proposed in this study is preferred to genetic algorithm proposed by Hassanzadeh et al. (2013) due to following reasons:

1 Figure 6 shows the convergence curve for Example 1. The curve shows convergence to the SP after 14 iterations of the existing genetic algorithm (Hassanzadeh et al., 2013) and after 6 iterations of the proposed PSO algorithm.

2 As can be seen from Table 2, the average number of iterations to converge for genetic algorithm is 6.8, while the average number of iterations to converge for genetic algorithm is 3.1.

3 As documented in Table 2, the minimum, maximum and average convergence time spans for genetic algorithm are 0.13, 1.43 and 0.61, respectively and these values for PSO algorithm are 0.11, 0.53 and 0.272. Due to this fact, utilising PSO algorithm is preferred to genetic algorithm for solving fuzzy SP from the time complexity point of view.

4 In addition, Table 2 gives the minimum, maximum and average total times to convergence using these algorithms. The minimum convergence total time among 10 times using the existing genetic algorithm and the proposed PSO algorithm are respectively 6.84 and 3.57. Also, the average convergence total time among 10 times using the existing genetic algorithm and the proposed PSO algorithm are respectively 8.718 and 4.726. Due to these facts, utilising PSO algorithm gives us a time advantage compared to genetic algorithm, regarding the convergence total time.

Example 5.2: Let us consider the network in Figure 5 with mixed fuzzy arc weights as given in Table 3. There are 23 nodes and 40 arcs in the network. Number of particles in the PSO algorithm and number of chromosomes in the genetic algorithm (Hassanzadeh et al., 2013) are 22. Number of iterations in both of these algorithms is 50.

Figure 5 The network for Example 2

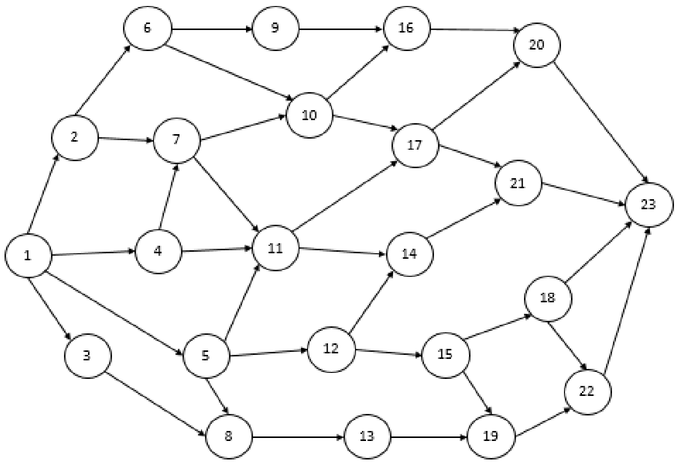


Table 3 The arc weights for Example 2

Arc	*Fuzzy number*	*Arc*	*Fuzzy number*	*Arc*	*Fuzzy number*
(1, 2)	(12, 13, 15, 17)	(1, 3)	(40, 11)	(1, 4)	(8, 10, 12, 13)
(1, 5)	(7, 8, 9, 10)	(2, 6)	(35, 10)	(2, 7)	(6, 11, 11, 13)
(3, 8)	(40, 11)	(4, 7)	(17, 20, 22, 24)	(4, 11)	(6, 10, 13, 14)
(5, 8)	(29, 9)	(5, 11)	(7, 10, 13, 14)	(5, 12)	(10, 13, 15, 17)
(6, 9)	(6, 8, 10, 11)	(6, 10)	(35, 11)	(7, 10)	(9, 10, 12, 13)
(7, 11)	(6, 7, 8, 9)	(8, 12)	(5, 8, 9, 10)	(8, 13)	(50, 5)
(9, 16)	(6, 7, 9, 10)	(10, 16)	(40, 13)	(10, 17)	(15, 19, 20, 21)
(11, 14)	(8, 9, 11, 13)	(11, 17)	(28, 9)	(12, 14)	(13, 14, 16, 18)
(12, 15)	(12, 14, 15, 16)	(13, 15)	(37, 12)	(13, 19)	(17, 18, 19, 20)
(14, 21)	(12, 12, 13, 14)	(15, 18)	(8, 9, 11, 13)	(15, 19)	(25,7)
(16, 20)	(38, 12)	(17, 20)	(7, 10, 11, 12)	(17, 21)	(6, 7, 8, 10)
(18, 21)	(15, 17, 18, 19)	(18, 22)	(16, 5)	(18, 23)	(15, 5)
(19, 22)	(5, 16, 17, 19)	(20, 23)	(13, 14, 16, 17)	(21, 23)	(12, 15, 17, 18)
(22, 23)	(20, 5)				

The corresponding fuzzy SP problem has been solved 10 times using the proposed PSO algorithm and the existing genetic algorithm (Hassanzadeh et al., 2013). The results are given in Table 4.

Table 4 Information corresponding to ten runs of Example 2

	Generation	*Shortest path*	*Number of iteration to converge*		*Convergence time span (s)*		*Total time (s)*	
			GA	*PSO*	*GA*	*PSO*	*GA*	*PSO*
1	80	1-3-8-7-11	5	3	1.94	0.80	23.72	18.89
2	80	1-3-8-7-11	2	2	1.09	0.66	23.73	18.49
3	80	1-3-8-7-11	4	1	1.45	0.51	23.80	18.76
4	80	1-3-8-7-11	3	1	1.28	0.51	23.69	18.27
5	80	1-3-8-7-11	11	8	3.02	2.13	23.91	18.78
6	120	1-3-8-7-11	15	1	4.62	0.53	35.34	27.01
7	120	1-3-8-7-11	2	2	0.75	0.63	35.54	27.53
8	120	1-3-8-7-11	3	1	1.46	0.50	35.30	26.55
9	120	1-3-8-7-11	5	7	1.64	1.21	35.35	27.22
10	120	1-3-8-7-11	13	1	4.11	0.51	35.44	27.68
Min	-	-	2	1	1.09	0.50	23.69	18.27
Max	-	-	15	8	4.62	2.13	35.54	27.68
Mean	-	-	6.3	2.7	2.136	0.799	29.582	18.318

Figure 6 Convergence curve of PSO and genetic algorithm for Example 1 (see online version for colours)

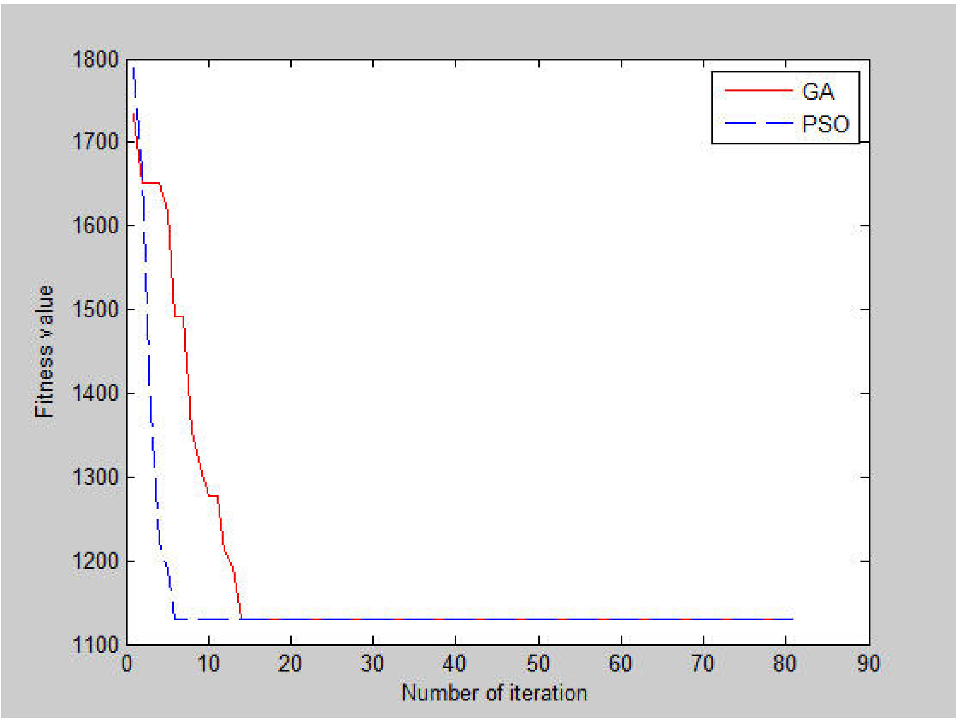
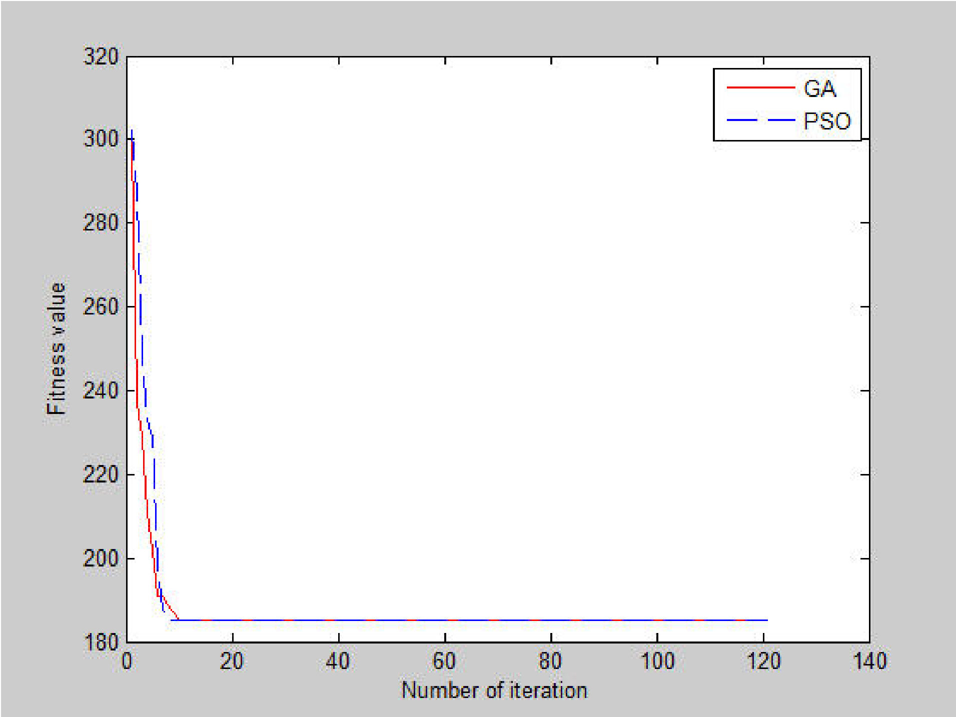


Figure 7 Convergence curve of PSO and Genetic Algorithm for Example 2 (see online version for colours)



After implementing our proposed approach, the fuzzy SP from node 1 to node 11 can be obtained: 1 → 5 → 12 → 15 → 18→23. The result is also consistent with the result in Hassanzadeh et al. (2013). However, using PSO algorithm proposed in this study for solving fuzzy SP is strongly economical compared to genetic algorithm proposed by Hassanzadeh et al. (2013) from the complexity time point of view due to following reasons:

1 Figure 7 shows the convergence curve for Example 2. The curve shows convergence to the SP after 15 iterations of the existing genetic algorithm (Hassanzadeh et al., 2013) and after 7 iterations of the proposed PSO algorithm.

2 As can be seen from Table 4, the minimum, maximum and average numbers of iterations to converge for genetic algorithm are 2, 15 and6.3 iterations, while the corresponding values for genetic algorithm are 1, 8 and 2.7 iterations.

3 As documented in Table 2, the average convergence time span for genetic algorithm is 2.136, while the corresponding time for PSO algorithm is 0.799.

4 In addition, From Table 2 the minimum convergence total time among 10 times using the existing genetic algorithm and the proposed PSO algorithm are respectively 23.69 and 18.27. Also, the average convergence total time among 10 times using the existing genetic algorithm and the proposed PSO algorithm are respectively 29.582 and 22.928. These facts confirm that utilising PSO algorithm gives us a time advantage compared to genetic algorithm, regarding the convergence total time.

6 Conclusions

In this paper, a solution procedure has been defined for a SP problem in which the network arcs are represented by mixed fuzzy numbers. We proposed a PSO algorithm for finding an optimal (shortest) path and its corresponding membership function instead of genetic algorithm. The comparative results illustrated that the algorithm proposed in this paper is more efficient than the existing genetic algorithm in terms of computing time even if the best result for each example is used. It should be noted that using the proposed algorithm in this study repeatedly for *n* (total number of nodes) times taking different source nodes, it is possible to find the SP between any pair of vertices. In future studies, we plan to develop the proposed approach to find the fuzzy shortest chain simultaneously between any two nodes considered. In addition, we plan to focus on developing the proposed algorithm to solve fuzzy SP problem on multi-weight networks. Finally, we plan to conduct further research by comparing the results obtained with those that might be obtained with other heuristic algorithms such as artificial bee colony (ABC) algorithm and artificial ant colony (ACO) algorithm. We hope that the concepts introduced here will provide inspiration for future research.

Acknowledgements

The authors would like to express their appreciation for the valuable comments made by three anonymous reviewers, which helped us to improve the quality of the paper.

References

Aboutahoun, A.W. (2010) 'Efficient solution generation for the bicriterion problems', *Int. J. of Operational Research*, Vol. 9, No. 3 pp.287–305.

Ahuja, R.K., Magnanti, T.L. and Orlin, J.B. (1993) *Network Flows: Theory, Algorithms and Applications*, Prentice-Hall, Englewood Cliffs, NJ.

Ardakani, M.K. and Tavana, M. (2015) 'A decremental approach with the *A** algorithm for speeding-up the optimization process in dynamic shortest path problems', *Measurement*, Vol. 60, pp.299–307.

Ardizzon, G., Cavazzini, G. and Pavesi, G. (2015) 'Adaptive acceleration coefficients for a new search diversification strategy in particle swarm optimization algorithms', *Information Sciences*, Vol. 299, pp.337–378.

Aridhi, S., Lacomme, P., Ren, L. and Vincent, B. (2015) 'A Map reduce-based approach for shortest path problem in large-scale networks', *Engineering Applications of Artificial Intelligence*, Vol. 41, pp.151–165.

Calazan, R.M., Nedjah, N. and Mourelle, L.M. (2014) 'A hardware accelerator for particle swarm optimization', *Applied Soft Computing*, Vol. 14, Part C, pp.347–356.

Chitra, C., Potti, S. and Subbaraj, P. (2011) 'A novel multi-objective evolutionary algorithm for shortest path routing problem', *Int. J. of Communication Networks and Distributed Systems*, Vol. 7, Nos. 3/4, pp.355–374.

Chuang, T.N. and Kung, J.Y. (2005) 'The fuzzy shortest path length and the corresponding shortest path in a network', *Computers and Operations Research*, Vol. 32, No. 6, pp.1409–1428.

Chuang, T.N. and Kung, J.Y. (2006) 'A new algorithm for the discrete fuzzy shortest path problem in a network', *Applied Mathematics and Computation*, Vol. 174, No. 1, pp.660–668.

Deng, Y., Chen, Y., Zhang, Y. and Mahadevan, S. (2012) 'Fuzzy Dijkstra algorithm for shortest path problem under uncertain environment', *Applied Soft Computing*, Vol. 12, No. 3, pp.1231–1237.

Dong, X., Li, W. and Zheng, L. (2013) 'Ant colony optimisation for a resource-constrained shortest path problem with applications in multimodal transport', *Int. J. of Modelling, Identification and Control*, Vol. 18, No. 3, pp.268–275.

Dou, Y., Zhu, L. and Wang, H.S. (2012) 'Solving the fuzzy shortest path problem using multi-criteria decision method based on vague similarity measure', *Applied Soft Computing*, Vol. 12, No. 6, pp.1621–1631.

Dubois, D. and Prade, H. (1980) *Fuzzy Sets and Systems: Theory and Applications*, Academic Press, New York.

Ebrahimnejad, A. (2011a) 'Sensitivity analysis in fuzzy number linear programming problems', *Mathematical and Computer Modelling*, Vol. 53, Nos. 9–10, pp.1878–1888.

Ebrahimnejad, A. (2011b) 'Some new results in linear programs with trapezoidal fuzzy numbers: Finite convergence of the Ganesan and Veeramani's method and a fuzzy revised simplex method', *Applied Mathematical Modelling*, Vol. 35, No. 9, pp.4526–4540.

Ebrahimnejad, A. (2013a) 'Some new results in linear programming problems with fuzzy cost coefficients', *Walailak Journal of Science and Technology*, Vol. 10, No. 2, pp.191–199.

Ebrahimnejad, A. (2013b) 'Tableau form of the fuzzy primal-dual simplex algorithm for solving linear programmes with trapezoidal fuzzy numbers', *International Journal of Operational Research*, Vol. 18, No. 2, pp.123–139.

Ebrahimnejad, A. (2014) 'A simplified new approach for solving fuzzy transportation problems with generalized trapezoidal fuzzy numbers', *Applied Soft Computing*, Vol. 19, pp.171–176.

Ebrahimnejad, A. and Tavana, M. (2014) 'A novel method for solving linear programming problems with symmetric trapezoidal fuzzy numbers', *Applied Mathematical Modelling*, Vol. 38, Nos. 17–18, pp.4388–4395.

Ebrahimnejad, A. and Verdegay, J.L. (2014a) 'A novel approach for sensitivity analysis in linear programs with trapezoidal fuzzy numbers', *Journal of Intelligent and Fuzzy Systems*, Vol. 27, No. 1, pp.173–185.

Ebrahimnejad, A. and Verdegay, J.L. (2014b) 'On solving bounded fuzzy variable linear program and its applications', *Journal of Intelligent and Fuzzy Systems*, Vol. 27, No. 5, pp.2265–2280.

Ebrahimnejad, A., Nasseri, S.H. and Mansourzadeh, S.M. (2013) 'Modified bounded dual network simplex algorithm for solving minimum cost flow problem with fuzzy costs based on ranking functions', *Journal of Intelligent and Fuzzy Systems*, Vol. 24, No. 1, pp.191–198.

Gao, H., Kwong, S., Yang, J. and Cao, J. (2013) 'Particle swarm optimization based on intermediate disturbance strategy algorithm and its application in multi-threshold image segmentation', *Information Sciences*, Vol. 250, pp.82–112.

Gao, Y. (2011) 'Shortest path problem with uncertain arc lengths', *Computers and Mathematics with Applications*, Vol. 62, No. 6, pp.2591–2600.

Grigoryan, H. and Harutyunyan, H.A. (2015) 'The shortest path problem in the Knödel graph', *Journal of Discrete Algorithms*, Vol. 31, pp.40–47.

Hassanzadeh, R., Mahdavi, I., Mahdavi-Amiri, N. and Tajdin, A. (2013) 'A genetic algorithm for solving fuzzy shortest path problems with mixed fuzzy arc lengths', *Mathematical and Computer Modelling*, Vol. 57, Nos. 1–2, pp.84–99.

Hernandes, F., Lamata, M.T., Verdegay, J.L. and Yamakami, A. (2007) 'The shortest path problem on networks with fuzzy parameters', *Fuzzy Sets and Systems*, Vol. 158, No. 14, pp.1561–1570.

Ji, X., Iwamura, K. and Shao, Z. (2007) 'New models for shortest path problem with problem with fuzzy arc lengths', *Applied Mathematical Modeling*, Vol. 31, No. 2, pp.259–269.

Kazemi, N., Ehsani, E. and Glock, C.H. (2014) 'Multi-objective supplier selection and order allocation under quantity discounts with fuzzy goals and fuzzy constraints', *Int. J. of Applied Decision Sciences*, Vol. 7, No. 1, pp.66–96.

Kennedy, J. and Eberhart, E. (1995) 'Particle swarm optimization', *IEEE International Conference on Neural Networks*, Vol. 4, pp.1942–1948.

Khalili-Damghani, K., Dideh-Khani, H. and Sadi-Nezhad, S. (2013) 'A two-stage approach based on ANFIS and fuzzy goal programming for supplier selection', *Int. J. of Applied Decision Sciences*, Vol. 6, No. 1 pp.1–14.

Klein, C.M. (1991) 'Fuzzy shortest paths', *Fuzzy Sets and Systems*, Vol. 39, No. 1, pp.27–41.

Kumar, A. and Kaur, M. (2011) 'A new algorithm, for solving shortest path problem on a network with imprecise edge weight', *Applications and Applied Mathematics: An International Journal*, Vol. 6, No. 2, pp.602–619.

Lai, C.N. (2015) On the construction of all shortest vertex-disjoint paths in Cayley graphs of Abelian groups', *Theoretical Computer Science*, Vol. 571, pp.10–20.

Li, Y., Gen, M. and Ida, K. (1996) 'Solving fuzzy shortest path problems by neural networks', *Computers and Industrial Engineering*, Vol. 31, Nos. 3–4, pp.861–865.

Lin, K.C. and Chern, M.S. (1993) 'The fuzzy shortest path problem and its most vital arcs', *Fuzzy Sets and Systems*, Vol. 58, No. 3, pp.343–353.

Lissovoi, A. and Witt, C. (2015) 'Runtime analysis of ant colony optimization on dynamic shortest path problems', *Theoretical Computer Science*, Vol. 561, No. 1, pp.73–85.

Mahdavi, I., Nourifar, R., Heidarzade, A. and Mahdavi-Amiri, N. (2009) 'A dynamic programming approach for finding shortest chains in fuzzy network', *Applied Soft Computing*, Vol. 9, No. 2, pp.503–511.

Mahi, M., Baykan, O.K. and Kodaz, H. (2015) 'A new hybrid method based on particle swarm optimization, ant colony optimization and 3-opt algorithms for traveling salesman problem', *Applied Soft Computing*, Vol. 30, pp.484–490.

Messaoudi, L. and Rebaï, A. (2014) 'A fuzzy goal programming approach for solving stochastic portfolio selection problem in a group decision-making context', *Int. J. of Applied Decision Sciences*, Vol. 7, No. 2, pp.208–221.

Moraes, A.O.S., Mitre, J.F., Lage, P.L.C. and Secchi, A.R. (2015) 'A robust parallel algorithm of the particle swarm optimization method for large dimensional engineering problems', *Applied Mathematical Modelling* [online] http://dx.doi.org/10.1016/j.apm.2014.12.034 (accessed 25 December 2014).

Nikulin, Y. and Iftikhar, Z. (2012) 'A note on different modelling approaches for the robust shortest path problem', *Int. J. of Mathematical Modelling and Numerical Optimisation*, Vol. 3, No. 3, pp.141–157.

Okada, S. (2004) 'Fuzzy shortest path problems incorporating interactivity among paths', *Fuzzy Sets and Systems*, Vol. 142, No. 3, pp.335–357.

Okada, S. and Soper, T. (2000) 'A shortest path problem on a network with fuzzy arc lengths', *Fuzzy Sets and Systems*, Vol. 109, No. 1, pp.129–140.

Olya, M.H. (2014a) 'Applying Dijkstra's algorithm for general shortest path problem with normal probability distribution arc length', *Int. J. of Operational Research*, Vol. 21, No. 2, pp.143–154.

Olya, M.H. (2014b) 'Finding shortest path in a combined exponential-gamma probability distribution arc length', *Int. J. of Operational Research*, Vol. 21, No. 1 pp.25–37.

Rangasamy, O., Akram, M. and Thilagavathi, S. (2013) 'Intuitionistic fuzzy shortest hyperpath in a network', *Information Processing Letters*, Vol. 113, No. 17, pp.599–603.

Rezvanian, A. and Meybodi, M.R. (2015) 'Sampling social networks using shortest paths', *Physica A: Statistical Mechanics and its Applications*, Vol. 424, pp.254–268.

Rudniy, A., Song, M. and Geller, J. (2010) 'Detecting duplicate biological entities using shortest path edit distance', *Int. J. of Data Mining and Bioinformatics*, Vol. 4, No. 4, pp.395–410.

Sadeghzadeh, H., Ehyaei, M.A. and Rosen, M.A. (2015) 'Techno-economic optimization of a shell and tube heat exchanger by genetic and particle swarm algorithms', *Energy Conversion and Management*, Vol. 93, pp.84–91.

Sahebjamnia, N., Yahiazadeh Andavari, Y., Safaie Koleti, Z. and Gholami Orimi, H. (2013) 'A possibilistic programming approach for capacitated lot-sizing problem in mixed assembly shops', *Int. J. of Applied Decision Sciences*, Vol. 6, No. 4, pp.388–405.

Shi, Y. and Eberhart, R. (1998) 'A modified particle swarm optimizer', *IEEE World Congress on Computational Intelligence*, pp.69–73.

Tajdin, A., Mahdavi, I., Mahdavi-Amiri, N. and Sadeghpour-Gildeh, B. (2010) 'Computing a fuzzy shortest path in a network with mixed fuzzy lengths using cut', *Computers and Mathematics with Applications*, Vol. 60, No. 2, pp.989–1002.

Tang, D., Cai, Y., Zhao, J. and Xue, Y. (2014) 'A quantum-behaved particle swarm optimization with memetic algorithm and memory for continuous nonlinear large scale problems', *Information Sciences*, Vol. 289, pp.162–189.

Trevizan, F.W. and Veloso, M.M. (2014) 'Depth-based short-sighted stochastic shortest path problems', *Artificial Intelligence*, Vol. 216, pp.179–205.

Yang, B. and Zhang, Q. (2010) 'Applying a modified particle swarm optimizer to discrete truss topology optimization', *International Conference on Computational Intelligence and Software Engineering (CiSE)*, pp.1–4.

Zhang, L., Tang, Y., Hua, C. and Guan, H. (2015) 'A new particle swarm optimization algorithm with adaptive inertia weight based on Bayesian techniques', *Applied Soft Computing*, Vol. 28, pp.138–149.

Zhang, Y., Zhang, Z., Deng, Y. and Mahadevan, S. (2013) 'A biologically inspired solution for fuzzy shortest path problems', *Applied Soft Computing*, Vol. 13, No. 5, pp.2356–2363.

Zhang, Z., Jiang, Y., Zhang, S., Geng, S., Wang, H. and Sang, G. (2014) 'An adaptive particle swarm optimization algorithm for reservoir operation optimization', *Applied Soft Computing*, Vol. 18, pp.167–177.