Measurement 93 (2016) 48–56

Contents lists available at ScienceDirect

Measurement

journal homepage: www.elsevier.com/locate/measurement

A novel artificial bee colony algorithm for shortest path problems with fuzzy arc weights

Ali Ebrahimnejad [a,*], Madjid Tavana [b,c], Hamidreza Alrezaamiri [d]

[a] Department of Mathematics, Qaemshahr Branch, Islamic Azad University, Qaemshahr, Iran
[b] Business Systems and Analytics Department, Distinguished Chair of Business Analytics, La Salle University, Philadelphia, PA 19141, USA
[c] Business Information Systems Department, Faculty of Business Administration and Economics, University of Paderborn, D-33098 Paderborn, Germany
[d] Young Researchers and Elite Club, Babol Branch, Islamic Azad University, Babol, Iran

ARTICLE INFO

Article history:
Received 30 September 2015
Received in revised form 3 May 2016
Accepted 22 June 2016
Available online 23 June 2016

Keywords:
Shortest path problem
Fuzzy numbers
Genetic algorithm
Particle swarm optimization
Artificial bee colony

ABSTRACT

The shortest path (SP) problem is a network optimization problem with a wide range of applications in business and engineering. Conventional network problems assume precise values for the weights of the edges. However, these weights are often vague and ambiguous in practical applications. Several heuristics have been proposed to find the shortest path (SP) weight and the corresponding SP on a network with fuzzy arc weights. These heuristics largely use α-cuts and the least squares method. We propose an artificial bee colony (ABC) algorithm to solve the fuzzy SP (FSP) problems with fuzzy arc weights. The performance of the proposed ABC algorithm is compared with the performance of other competing algorithms with two SP problems taken from the literature. We present a wireless sensor network (WSN) problem and demonstrate the applicability of the proposed method and exhibit the efficiency of the procedures and algorithms.

1. Introduction

The shortest path (SP) problem is one of the most studied problems in the fields of combinatorics and network optimization. The aim of the SP problem is to find a path between two specified nodes and optimize the weight (cost, time or distance) of the path. Traditional SP problems assume the arc weights can be specified precisely. However, the weights of the arcs in most real-life problems are imprecise and ambiguous. For example, arc weights that are represented by time or cost can fluctuate with traffic conditions, weather, or payload. In such cases, fuzzy numbers can be used to represent the imprecision in the arc weights. SP problems with fuzzy numbers are called fuzzy SP (FSP) problems. Much research has been carried out on FSP modeling and solution procedures.

Dubois and Prade [10] were among the first researchers who extended the classic Floyd and Ford–Moore–Bellman algorithms to solve FSP problems. Klein [21] discussed the possibility that the FSP may not correspond to an actual path in the network and proposed new models based on dynamic programming to avoid this problem. Lin and Chern [23] proposed an algorithm for finding the single most vital arc in a network as being the one whose removal from the path results in the greatest increase in cost. Okada and Soper [27] proposed an algorithm to obtain all Pareto optimal paths from a specific node to every other node in a network with fuzzy numbers. Okada [26] developed an algorithm to determine the degree of possibility for each arc on the SP. Chuang and Kung [4] proposed a heuristic procedure to find the FSP length among all possible paths in a network. Chuang and Kung [5] proposed a new algorithm that obtains the FSP length and the corresponding SP in a discrete FSP problem. Hernandes et al. [15] considered a genetic algorithm (GA) for solving FSP problems where the decision maker can choose a ranking index that best suits the problem. Ji et al. [18] proposed a hybrid intelligent algorithm integrating simulation and the GA to solve three types of models for the FSP problem in a fuzzy environment. Mahdavi et al. [25] proposed a dynamic programming approach to solve the shortest chain problem with fuzzy distances for every arc using a ranking method. Kumar and Kaur [22] presented a new algorithm for solving the SP problem on a network with imprecise arc weights. Dou et al. [9] applied an approach to select the SP in a multi-constrained network using a multi-criteria decision making method based on a vague similarity measure. Deng et al. [8] extended the Dijkstra algorithm to solve the SP problem with fuzzy

* Corresponding author.
E-mail addresses: aemarzoun@gmail.com, a.ebrahimnejad@qaemiau.ac.ir (A. Ebrahimnejad), tavana@lasalle.edu (M. Tavana), hamidreza.alreza@baboliau.ac.ir (H. Alrezaamiri).
URL: http://tavana.us/ (M. Tavana).

http://dx.doi.org/10.1016/j.measurement.2016.06.050
0263-2241/

arc weights. Their proposed method is based on the graded mean integration representation of fuzzy numbers. Zhang et al. [35] proposed a biologically inspired algorithm called the fuzzy physarum algorithm for FSP problems based on a path finding model.

These studies assume the arc weights of the network under consideration have the same type of fuzzy numbers. To overcome this limitation, Tajdin et al. [31] designed an algorithm for computing a SP in a network with various types of fuzzy arc weights. They used an α-cut approach to compute the addition of various fuzzy numbers as arc weights. Hassanzadeh et al. [14] presented a GA for finding the SP in the network to overcome the complexity of the addition of various fuzzy numbers encountered in larger problems. Ebrahimnejad et al. [11] used a population-based metaheuristic optimization algorithm, namely particle swarm optimization (PSO), to approximate the SP on the same network, where arcs are weighted with different types of fuzzy numbers. In this study, we design an artificial bee colony (ABC) algorithm to solve the FSP in the same network using a recently proposed distance function for comparison of fuzzy numbers. The use of evolutionary computation techniques and algorithms such as the ABC is increasing in different measurement applications, due to their capacity to operate within complex environments and provide accurate solutions to the optimization problem being considered [3,29,30]. In this regard, it will be shown that the proposed method significantly reduces the complexities encountered in the existing methods. In addition, we demonstrate the applicability of the proposed approach and exhibit the efficiency of the procedures and algorithms in a wireless sensor network (WSN) problem.

The remainder of this paper is structured as follows. In Section 2, basic concepts and definitions of fuzzy set theory, α-cut computations for fuzzy numbers, and the distance between fuzzy numbers are reviewed. The proposed ABC algorithm for solving the FSP problem is presented in Section 3. The performance of the proposed heuristic algorithm is investigated on two different SP problems in Section 4. The application of the proposed method in a WSN problem is presented in Section 5. In Section 6, we introduce other practical applications of the FSP networks and in Section 6, we outline our conclusions and future research directions.

2. Preliminaries

In this section, we present some basic definitions and arithmetic operations on fuzzy numbers [10,31,14].

Definition 1. Let the universal set be X and define the fuzzy set $\widetilde{A}$ in X by its membership function $\mu_{\widetilde{A}} : X \to [0,1]$, where a real number $\mu_{\widetilde{A}}(x)$ is assigned to each element $x \in X$ in the interval [0, 1].

Definition 2. The α-cut of a fuzzy set $\widetilde{A}$ is defined as a crisp set $[\widetilde{A}]_\alpha$ in which the membership degrees of its elements exceed the level α, i.e. $[\widetilde{A}]_\alpha = \{x \in X; \mu_{\widetilde{A}}(x) \geqslant \alpha\} = \left[\widetilde{A}^L_\alpha, \widetilde{A}^R_\alpha\right]$.

Definition 3. A fuzzy number is a convex normalized fuzzy set of the real line R, whose membership function is piecewise continuous.

Definition 4. A normal fuzzy number is represented by $\widetilde{A} = (m, \sigma)$, with the membership function, $\mu_{\widetilde{A}}$, defined by the expression

$$\mu_{\widetilde{A}}(x) = e^{-\left(\frac{x-m}{\sigma}\right)^2}, \; x \in R \tag{1}$$

where m is the mean and σ is the standard deviation (see Fig. 1).

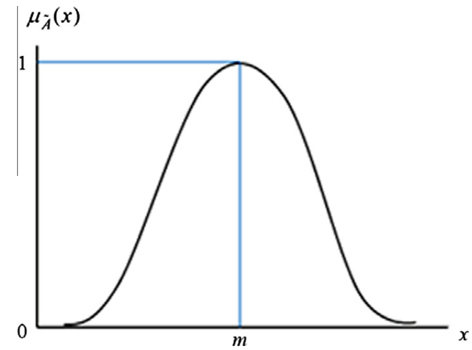


Fig. 1. A normal fuzzy number $\widetilde{A} = (m, \sigma)$.

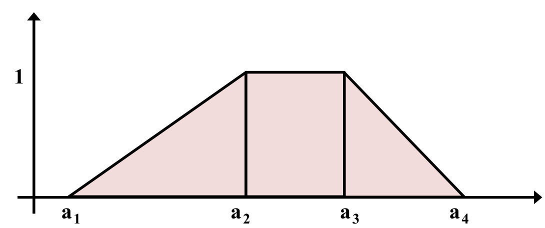


Fig. 2. A trapezoidal fuzzy number $\widetilde{A} = (a_1, a_2, a_3, a_4)$.

Definition 5. The α-cut of normal fuzzy number $\widetilde{A} = (m, \sigma)$ is given by $[\widetilde{A}]_\alpha = \left[\widetilde{A}^L_\alpha, \widetilde{A}^R_\alpha\right] = \left[m - \sigma\sqrt{-\ln(\alpha)}, m + \sigma\sqrt{-\ln(\alpha)}\right]$.

Definition 6. A trapezoidal fuzzy number $\widetilde{A}$ is represented by $\widetilde{A} = (a_1, a_2, a_3, a_4)$, with the following membership function (see Fig. 2):

$$\mu_{\widetilde{A}}(x) = \begin{cases} \frac{x-a_1}{a_2-a_1}, & a_1 \leqslant x \leqslant a_2, \\ 1, & a_2 \leqslant x \leqslant a_3, \\ \frac{a_4-x}{a_4-a_3}, & a_3 \leqslant x \leqslant a_4. \end{cases} \tag{2}$$

Definition 7. The α-cut of the trapezoidal fuzzy number $\widetilde{A} = (a_1, a_2, a_3, a_4)$ is given by $[\widetilde{A}]_\alpha = \left[\widetilde{A}^L_\alpha, \widetilde{A}^R_\alpha\right] = [(a_2 - a_1)\alpha + a_1, a_4 - (a_4 - a_3)\alpha]$.

A key issue in SP problems with different types of fuzzy arc weights is the addition of trapezoidal and normal fuzzy numbers. Hassanzadeh et al. [14] proposed a step-by-step procedure for approximating the sum of trapezoidal and normal fuzzy numbers. In this procedure, Hassanzadeh et al. [14] approximate the sum and its corresponding membership function by dividing the α-interval, [0, 1], into n subintervals and letting $\alpha_0 = 0$; $\alpha_i = \alpha_{i-1} + \Delta\alpha_i$; $\Delta\alpha_i = \frac{1}{n}$; and $n = 1, 2, \ldots, n$.

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

$$\begin{aligned}[\widetilde{C}]_{\alpha_i} &= \left[\widetilde{C}^L_{\alpha_i}, \widetilde{C}^R_{\alpha_i}\right] = \left[\widetilde{A}^L_{\alpha_i} + \widetilde{B}^L_{\alpha_i}, \widetilde{A}^R_{\alpha_i} + \widetilde{B}^R_{\alpha_i}\right] \\ &= \Big[(a_2 - a_1)\alpha_i + a_1 + m - \sigma\sqrt{-\ln(\alpha_i)}, a_4 - (a_4 - a_3)\alpha_i \\ &\quad + m + \sigma\sqrt{-\ln(\alpha_i)}\Big]\end{aligned} \tag{3}$$

Eq. (3) can be used to obtain n points for $\widetilde{C}^L_{\alpha_i}$ and n points for $\widetilde{C}^R_{\alpha_i}$ using $\alpha_i, 1 \leqslant i \leqslant n$.

Hassanzadeh et al. [14] approximated the membership function of the sum using the resulting points via the α –cut and Crammer's approach for fitting an exponential membership function for the sum. Let $x_i = \widetilde{C}^R_{\alpha_i}$ and $y_i = \mu(\widetilde{C}^R_{\alpha_i})$, and for n points (x_i, y_i), consider the fitting model to be $y = e^{-\left(\frac{x-\lambda}{\beta}\right)^2}$. 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 [31,14]:

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

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

Similarly, let $x_i = \widetilde{C}^L_{\alpha_i}$ and $y_i = \mu\left(\widetilde{C}^L_{\alpha_i}\right)$, and consider the fitting model $y = e^{-\left(\frac{x-\lambda'}{\beta'}\right)^2}$. The least squares model for approximating the left membership function of the considered addition results in the unknown parameters λ' and β' as follows [31,14]:

$$\beta' = \frac{n\sum_i\left(x_i \times \sqrt{-\ln y_i}\right) - \sum_i\sqrt{-\ln y_i} \times \sum_i x_i}{n\sum_i\sqrt{\ln y_i} + \sum_i\sqrt{-\ln y_i} \times \sum_i\sqrt{-\ln y_i}} \tag{6}$$

$$\lambda' = \frac{\sum_i \ln y_i \times \sum_i x_i + \sum_i(x_i \times \sqrt{-\ln y_i}) \times \sum_i\sqrt{-\ln y_i}}{n\sum_i\sqrt{\ln y_i} + \sum_i\sqrt{-\ln y_i} \times \sum_i\sqrt{-\ln y_i}} \tag{7}$$

Therefore, the approximate membership function for the approximating sum of trapezoidal and normal fuzzy numbers is given as follows:

$$\mu_{\tilde{c}}(x) = \begin{cases} e^{-\left(\frac{\lambda'-x}{\beta'}\right)^2}, & x < \lambda', \\ 1, & \lambda' \leqslant x \leqslant \lambda, \\ e^{-\left(\frac{x-\lambda}{\beta}\right)^2}, & x > \lambda. \end{cases} \tag{8}$$

The other key issue in the SP problem is comparing the distance between two different paths with mixed fuzzy arc weights. In what follows, the distance between two fuzzy numbers using the resulting points from the α-cut is reviewed [31,14].

Given two fuzzy numbers $\widetilde{A}$ and $\widetilde{B}$, $D_{p,q}$, the distance between them is defined as follows:

$$D_{p,q}(\widetilde{A},\widetilde{B}) = \begin{cases} \left[(1-q)\int_0^1 |A^-_\alpha - B^-_\alpha|^p d\alpha + q\int_0^1 |A^+_\alpha - B^+_\alpha|^p d\alpha\right], & p < \infty \\ (1-q)\sup\limits_{0<\alpha\leqslant 1}|A^-_\alpha - B^-_\alpha| + q\inf\limits_{0<\alpha\leqslant 1}|A^+_\alpha - B^+_\alpha|, & p = \infty \end{cases} \tag{9}$$

where the first parameter p denotes the priority weight assigned 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 $\widetilde{A}$ and $\widetilde{B}$, $D_{p,q}$ is proportional to:

$$D_{p,q}(\widetilde{A},\widetilde{B}) = \left[(1-q)\sum_{i=1}^n\left|A^-_{\alpha_i} - B^-_{\alpha_i}\right|^p + q\sum_{i=1}^n\left|A^+_{\alpha_i} - B^+_{\alpha_i}\right|^p\right]^{\frac{1}{p}} \tag{10}$$

If $q = \frac{1}{2}$ and $p = 2$, we obtain the following:

$$D_{2,\frac{1}{2}}(\widetilde{A},\widetilde{B}) = \sqrt{\left[\frac{1}{2}\sum_{i=1}^n\left|A^-_{\alpha_i} - B^-_{\alpha_i}\right|^2 + \frac{1}{2}\sum_{i=1}^n\left|A^+_{\alpha_i} - B^+_{\alpha_i}\right|^2\right]} \tag{11}$$

To compare two fuzzy arc weights $\widetilde{A}$ and $\widetilde{B}$ using the α_i-cuts, we compare them to $\tilde{0} = (0, 0, \ldots, 0)$ because they are supposed to represent positive values. In fact, Eq. (9) is used to compute $D_{2,\frac{1}{2}}(\widetilde{A},\tilde{0})$ and $D_{2,\frac{1}{2}}(\widetilde{B},\tilde{0})$. In this case, we can conclude that $\widetilde{A} \preceq \widetilde{B}$ if and if $D_{2,\frac{1}{2}}(\widetilde{A},\tilde{0}) \leqslant D_{2,\frac{1}{2}}(\widetilde{B},\tilde{0})$.

3. ABC algorithm for finding FSP

A large number of heuristic algorithms have been proposed in the literature to solve combinatorial optimization problems [2,24]. Some of these algorithms are designed to solve the SP problems with different types of fuzzy arc weights. Hassanzadeh et al. [14] proposed a GA for solving SP problems with different types of fuzzy arc weights where the addition of various types of fuzzy numbers adds significantly to the computational complexities of large problems. Ebrahimnejad et al. [11] proposed a PSO algorithm for finding the SP in a network where the arcs are weighted with different types of fuzzy numbers.

3.1. Artificial bee colony algorithm

The ABC algorithm was proposed by Karaboga [19] for solving numerical optimization problems. This simple and powerful algorithm is motivated from the foraging behavior of honey bee swarms. In this algorithm, the bees are divided into three categories: the employed bees, the onlooker bees and the scouts. The employed bees search for the food source. The onlookers are those who are waiting in the dance area for getting information and making a decision whether to go towards a food source or not. The scout bees look for finding a food source in the search area randomly. Every food source is a solution in the optimization problem and also the amount of nectar in the food source is considered as a fitness solution. The process begins with the algorithm generating the initial population with random values of (*SN*, *D*) where *SN* is the number of members in the population and *D* is the dimension of each problem solution. After initializing the population, the algorithm starts the search process for each bee in a $C = 1, 2, \ldots, MCN$ loop where *MCN* is the number of iterations. In other words, the algorithm first performs the employed bees' actions in each iteration and then depending on the results of their actions, it then performs the actions of onlookers' bees, and finally it performs the actions of scout bees if necessary.

In this algorithm, the first half of the colony consists of employed bees and the second half constitutes the onlookers. The number of the employed bees or the onlooker bees is equal to the number of solutions in the population. In each iteration every employed bee performs the local search around their corresponding food source and if they find a better food source, they work on a new food source and ignore the previous one. Meanwhile, they ignore the new food source if it has less fitness than the previous source. The employed bees share the information about the amount of fitness and the location of the corresponding food source with onlooker bees after they finish their work. Every onlooker bee chooses a food source for doing a local search. The selection probability of each food source correlates with the amount of fitness. This means that the greater the value of fitness in the food source, the greater the probability of being selected by the onlookers. The onlooker bee performs a local search after choosing a food source hoping to find a better food source with more fitness. The ABC algorithm uses a predetermined value called the "*limit*" for an employed bee in order not to be constrained in utilizing a permanent food source. This means that if a bee couldn't improve the food source in the local search in several successive iterations which is determined by the *limit* variable, it should leave that food source and the employed bee becomes a scout bee.

The scout bee's task is to find a new food source by randomly exploring the problem area. After they finish their work, the algorithm moves on to the next iteration and this procedure continues until the number of iterations equals to MCN. The performance of the ABC algorithm is dependent on its local searches, the greedy selection of employed and onlooker bees, and the global search done by the onlookers. There are three important control parameters in the ABC algorithm: the number of food sources (*SN*) that is equal to the number of employed and onlooker bees, the determinative limitation that defines the maximum number of unsuccessful successive local searches for leaving a food source, and the number of iterations of the algorithm which equals to MCN [20].

3.2. Finding the SP by ABC

In this section we explain how to generate the initial population, the activity of the employed, onlooker and scout bees and how to calculate the fitness.

3.2.1. ABC population initialization

In order to find the SP using the ABC algorithm, an initial population is established. The initial population consists of solutions that correspond to paths from the origin of the graph to the destination of the graph. For establishing the path in a graph, the vicinity matrix to that graph is required. First the vicinity matrix is established for each graph and then the paths for the initial population are established [14]. A simple graph and its vicinity matrix are shown in Fig. 3. Since the lengths of the paths are not equal, the lengths of the solutions are also not equal. Fig. 4 shows two paths in the graph of Fig. 3 that can be considered as a solution.

Algorithm 1: Producing the initial population
(1) Determine the vicinity matrix of directed network $G = (V, E)$, give $Solution - size$ and set $q = 1$.
(2) Set $i = 1$, $m = 1$ and $p(m) = 1$.
(3) Define $a^1(i) = \{j|(i,j) \in A, a_{ij} = 1\}$ 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 labeling vector p. Let $q = q + 1$.
(6) If $q \leqslant Solution - size$ then go to (2) else stop.

3.2.2. Performance of employed bees

The number of employed bees in the ABC algorithm is equal to number of solutions of the population. In addition, the employed bee i is related to solution i in the population. In each iteration,

	1	3	5	7
1	4	6	5	7

Fig. 4. Two network paths.

each employed bee must make a local search around the corresponding solution. Here, the mutation operator in the GA is used as local search. If the fitness value of the new path results in a mutation that is more than that of the current path, the bee uses the new path and ignores the previous one. Otherwise, the bee forgets the new path.

A heuristic approach is used to do a local search since the SP problem is a discrete optimization problem and consists of a number of specified paths. This means that the mutation operator is used as a local search. Local search is defined as the "random change of path from the middle of the path to the destination." In other words, with random selection of one of the intermediate nodes of a path, the path will shift to the destination point. Thus, the new path is identical to the previous path from the origin node until the node is randomly selected.

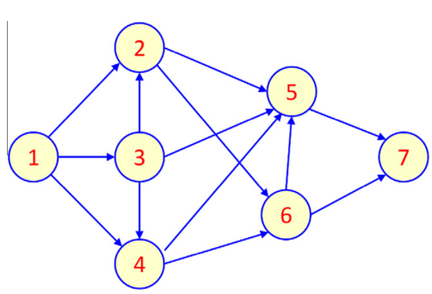
It should be noted that in the GA proposed by Hassanzadeh et al. [14], the number of paths for the mutation operator is determined using a mutation operator's rate or probability (p_m). However, the ABC algorithm proposed in this study doesn't use this variable because each worker bee should have a local search (mutation operator) on its solution and so the mutation operator is required on each path.

3.2.3. Performance of onlooker bees

When the employed bees have completed the search process, they share the obtained information including the nectar amount of each source and its position (in this algorithm each path and its fitness value) with onlooker bees in the dance area. Each onlooker bee then chooses one of the paths depending on the probability associated with the fitness value of each path by using a selection method such as the Roulette Wheel. The probability of each selection (p_i) is given as follows [20]:

$$p_i = \frac{fit_i}{\sum_{n=1}^{SN} fit_n} \tag{12}$$

where fit_i is the fitness amount of path i and SN is the number of paths of the population which is equal to the number of employed bees. The path with a higher fitness value has a greater probability to be selected by the onlooker bees. After that, each onlooker bee selects a path, and does a local search on this path. Recall that we use the mutation operator as a local search. If the new path obtained by the mutation operator has a higher fitness value than the old one, the algorithm forgets the old one and changes the number of unsuccessful local searches associated with the new path to



$$A=\begin{bmatrix} 0&1&1&1&0&0&0 \\ 0&0&0&0&1&1&0 \\ 0&1&0&1&1&0&0 \\ 0&0&0&0&1&1&0 \\ 0&0&0&0&0&0&1 \\ 0&0&0&0&1&0&1 \end{bmatrix}$$

Fig. 3. A simple network with its vicinity matrix.

zero. Otherwise, if the new path obtained by the mutation operator has a lower fitness value than the previous one, the algorithm forgets the new path and then the number of unsuccessful local searches associated with the new path increases by one unit.

3.2.4. Performance of scout bees

In the ABC algorithm, providing that a food source position (a path) cannot be improved further through a predetermined number of cycles, that food source (that path) is then assumed to be abandoned. The number of successive failed iterations is called the *limit* for abandonment. For those paths whose failed iterations are higher than their limits, the corresponding employed bees become scout bees. Scout bees forget their previous path by moving in the problem space and generating a new random path. A new path is then obtained by Algorithm 1 and is replaced with the previous path in the solution.

3.2.5. Evaluating fitness

The process of fitness evaluation of each solution is the same in the extended GA [14], in the extended PSO algorithm [11], and in the extended ABC algorithm proposed in this study. This means that this value is found by aggregating the arcs included in the path, where Eq. (3) is applied to sum the various arcs. The result of the addition is a set of α-cut points. Then, in order to compare the paths, the distance function $D_{2,\frac{1}{2}}$ is used. The values of $D_{2,\frac{1}{2}}$ is the path length and the minimum possible value in the network is the SP length [14]. The flowchart illustrating the procedure of the proposed algorithm is shown in Fig. 5.

4. Comparative examples

We solve the numerical examples given in Hassanzadeh et al. [14] and Ebrahimnejad et al. [11] with the algorithm proposed in this study for comparison purposes.

Example 1. Fig. 6 shows a network with 11 nodes and 25 arcs having different types of fuzzy weights as given in Table 1. The number of solutions of the population of the ABC algorithm is 10. In order to make a fair comparison, the number of solutions of the ABC algorithm is chosen to be the same as the number of particles used in the PSO algorithm and the number of chromosomes used in the GA. The number of iterations in all the algorithms is identical.

The corresponding FSP problem is solved 10 times using the proposed ABC algorithm, the existing PSO algorithm [11] and the existing GA [14]. The results are given in Table 2.

The proposed ABC algorithm finds the path $1 \rightarrow 3 \rightarrow 8 \rightarrow 7 \rightarrow 11$ as the SP from node 1 to node 11 matching the results of Hassanzadeh et al. [14] and Ebrahimnejad et al. [11]. We cite the following reasons to show that using the ABC algorithm proposed in this study is preferred to the GA proposed by Hassanzadeh et al. [14] and the PSO algorithm proposed by Ebrahimnejad et al. [11]:

(a) The convergence curve for Example 1 is shown in Fig. 7. The curve shows convergence to the SP after 12 iterations of the existing GA, after 7 iterations of the existing PSO algorithm, and after 5 iterations of the proposed ABC algorithm. The number of iterations of the proposed ABC algorithm to convergence is less than that of the GA and the PSO algorithm.

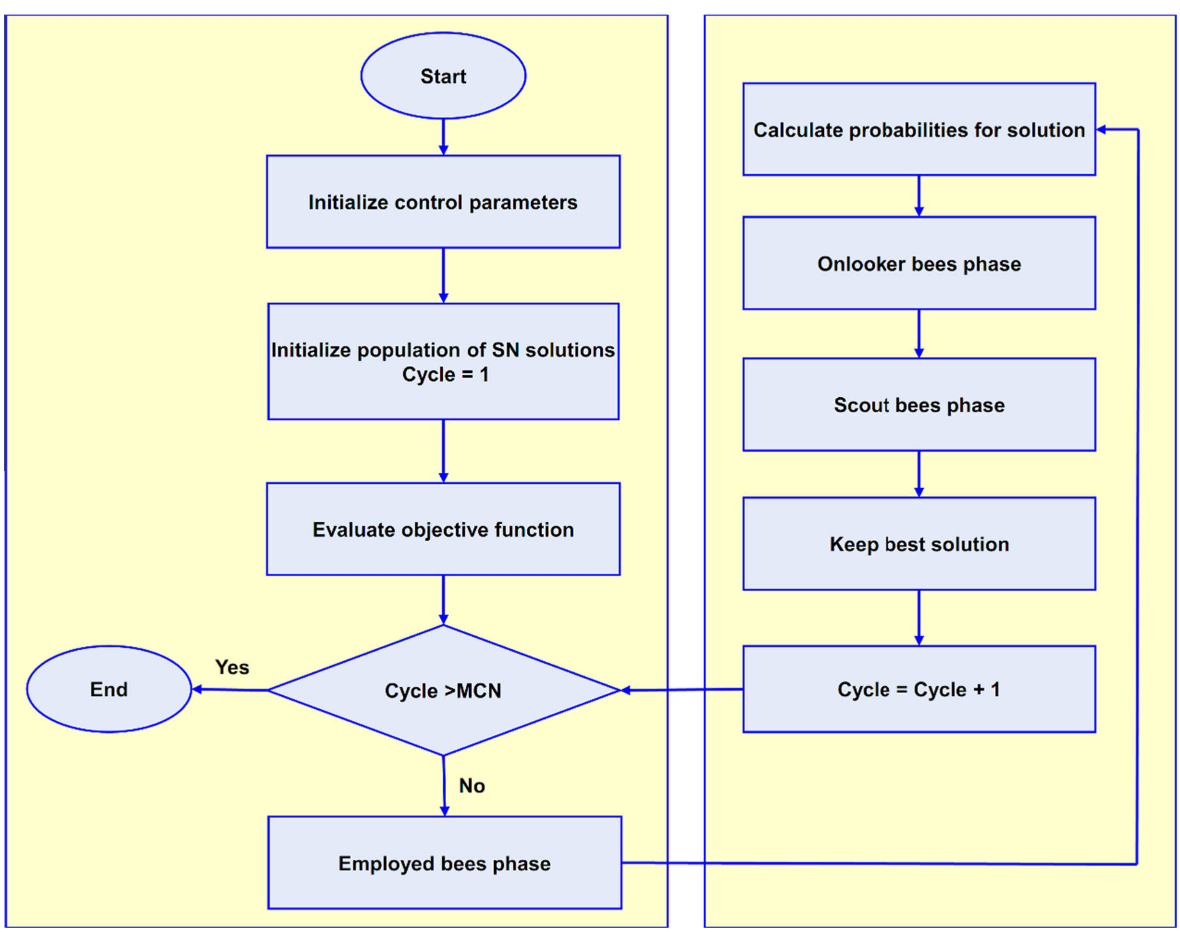


Fig. 5. Proposed algorithm flowchart.

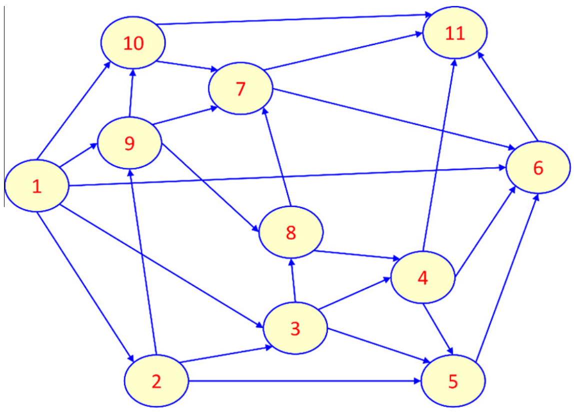


Fig. 6. The network for Example 1.

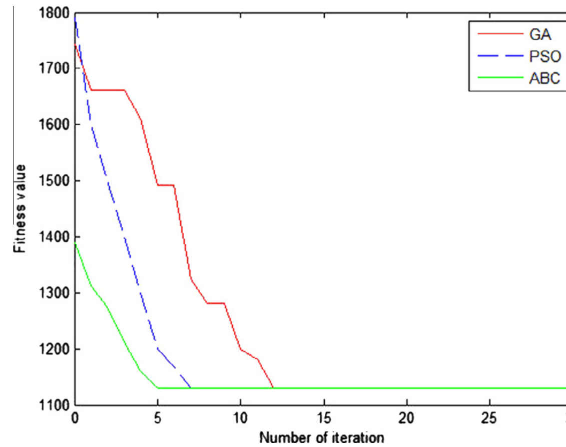


Fig. 7. Convergence curve of genetic algorithm, PSO and ABC algorithms for Example 1.

(b) Table 2 shows the average numbers of iterations to converge for the GA and the PSO algorithm are 7.1 and 4.0, respectively, while the average number of iterations to converge for the ABC algorithm is 3.1.
(c) As documented in Table 2, the minimum, maximum and average convergence time spans for the GA are 0.18, 1.55 and 0.75, respectively. These values for the PSO algorithm are 0.09, 0.54 and 0.29, while the corresponding values for the ABC algorithm are 0.6, 0.33 and 0.16. Thus, the ABC algorithm is preferred to the GA and the PSO algorithms for finding FSP in terms of time.
(d) In addition, the minimum, maximum and average total times to convergence using these algorithms are given in Table 2. The minimum convergence total time for the 10 cases using the existing GA, the existing PSO algorithm and the proposed ABC algorithm are respectively 2.54, 1.58 and 1.56. Also, the average convergence total time for the 10 cases using the existing GA, the existing PSO algorithm and the proposed ABC algorithm are respectively 2.91, 1.69 and 1.66. Thus, the ABC algorithm gives a total convergence time advantage compared to the genetic and PSO algorithm.

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)	(7, 6)	(390, 410, 540)	(3, 4)	(650, 667, 983)
(7, 11)	(45, 15)				

Example 2. Let us consider the network in Fig. 8 with different types of fuzzy arc weights as given in Table 3. There are 23 nodes and 40 arcs in the network. The number of solutions of the ABC algorithm, the number of particles in the PSO algorithm [11], and the number of chromosomes in the GA [14] are 22. The number of iterations in both algorithms is identical.

The corresponding FSP problem has been solved 10 times using the proposed ABC algorithm, the existing GA [14], and the existing PSO algorithm [11]. The results are given in Table 4.

After implementing the proposed ABC algorithm, the FSP from node 1 to node 11 is obtained as follows: 1 → 5 → 12 → 15 → 18 → 23. The result is also consistent with the results in Hassanzadeh et al. [14] and Ebrahimnejad et al. [11]. However, using the ABC algorithm for solving the FSP problem is preferred to the GA proposed by Hassanzadeh et al. [14] and the PSO algorithm proposed by Ebrahimnejad et al. [11] in terms of time complexity due to the following reasons:

Table 2
Information corresponding to ten runs of Example 1.

	Generation	SP	Number of iteration to converge			Convergence time span (s)			Total time (s)		
			GA	PSO	ABC	GA	PSO	ABC	GA	PSO	ABC
1	30	1–3–8–7–11	5	4	4	0.55	0.28	0.21	2.54	1.69	1.57
2	30	1–3–8–7–11	9	2	4	0.99	0.17	0.22	3.02	1.77	1.68
3	30	1–3–8–7–11	3	5	2	0.28	0.33	0.09	2.81	1.67	1.61
4	30	1–3–8–7–11	15	4	2	1.55	0.25	0.09	2.64	1.68	1.59
5	30	1–3–8–7–11	6	8	1	0.59	0.54	0.08	3.06	1.77	1.56
6	30	1–3–8–7–11	2	1	3	0.26	0.09	0.17	3.00	1.68	1.66
7	30	1–3–8–7–11	1	2	4	0.18	0.18	0.19	3.01	1.67	1.56
8	30	1–3–8–7–11	9	6	1	0.93	0.37	0.06	2.94	1.75	1.88
9	30	1–3–8–7–11	10	7	3	1.09	0.48	0.33	3.07	1.63	1.75
10	30	1–3–8–7–11	11	3	7	1.11	0.22	0.19	3.02	1.58	1.69
Min	–	–	1	1	1	0.18	0.09	0.06	2.54	1.58	1.56
Max	–	–	15	8	7	1.55	0.54	0.33	3.07	1.77	1.88
Mean	–	–	7.1	4	3.10	0.75	0.29	0.16	2.91	1.69	1.66

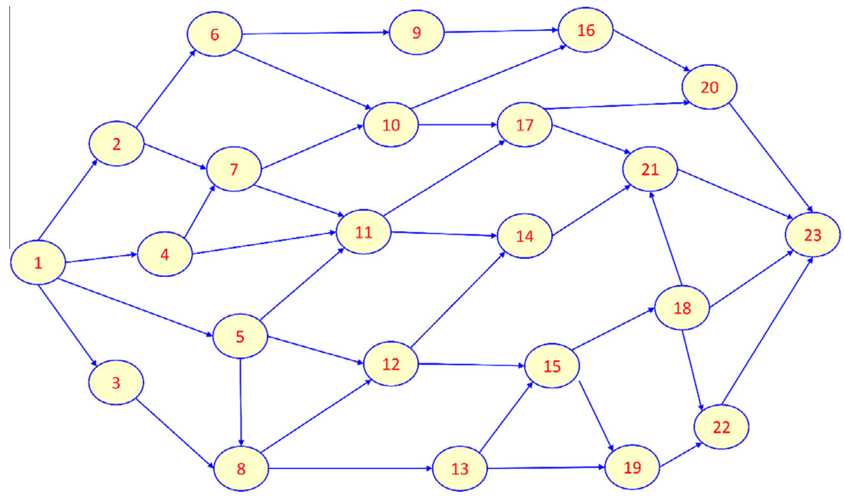


Fig. 8. The network for Example 2.

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)				

(a) Fig. 9 shows the convergence curve for Example 2. The curve shows convergence to the SP after 10 iterations of the existing GA, after 8 iterations of the existing PSO algorithm and after 5 iterations of the proposed ABC algorithm.
(b) As shown in Table 4, the average numbers of iterations to converge for the GA and the PSO algorithm are 7.6 and 2.7 iterations, while this value for the ABC algorithm is 2.6 iterations.
(c) As documented in Table 4, the average convergence time spans for the GA and the PSO algorithm are 2.13 and 1.36, respectively, while the corresponding time for the ABC algorithm is 0.95.
(d) Table 4 shows the average time for 10 cases using the existing GA and the existing PSO algorithm are respectively 9.5 and 7.39, while this value is only 5.39 for the ABC algorithm. Thus the ABC algorithm gives a total convergence time advantage over the GA and the PSO algorithm.

5. Application in wireless sensor networks

WSNs are being utilized increasingly in critical applications and consequently many researchers have chosen to study this area [7,13,17,34]. A WSN consists of a number of self-powered devices that can sense and communicate with other devices for the purpose of gathering local information to make global decisions about a physical environment [28]. Data gathered may include a variety of environmental conditions such as temperature, humidity, pressure, and early fire detection [16]. The most restrictive constraint imposed by these networks is energy sources. The energy source, which limits the lifetime of the network, has received considerable attention by researchers in recent years. Energy aware protocols, which are designed to save as much energy as possible, extend the lifetime of the network [32].

Table 4
Information corresponding to ten runs of Example 2.

	Generation	SP	Number of iteration to converge			Convergence time span (s)			Total time (s)		
			GA	PSO	ABC	GA	PSO	ABC	GA	PSO	ABC
1	30	1–5–12–15–18–23	5	4	2	1.76	1.80	0.69	9.19	7.44	5.33
2	30	1–5–12–15–18–23	3	2	3	1.61	1.00	0.98	9.63	7.33	5.26
3	30	1–5–12–15–18–23	8	3	1	2.15	1.41	0.46	9.88	7.41	5.54
4	30	1–5–12–15–18–23	4	1	5	1.68	0.51	1.97	9.55	7.29	5.41
5	30	1–5–12–15–18–23	2	5	3	1.35	1.91	1.03	9.42	7.65	5.28
6	30	1–5–12–15–18–23	10	1	1	2.88	0.55	0.47	9.81	7.31	5.21
7	30	1–5–12–15–18–23	17	8	2	4.87	2.12	0.63	9.34	7.55	5.61
8	30	1–5–12–15–18–23	12	4	1	3.13	1.78	0.41	9.30	7.34	5.42
9	30	1–5–12–15–18–23	9	6	6	2.64	1.97	2.14	9.35	7.11	5.51
10	30	1–5–12–15–18–23	6	1	2	2.11	0.52	0.68	9.51	7.47	5.31
Min	–	–	2	1	1	1.09	0.51	0.41	9.19	7.11	5.21
Max	–	–	17	8	6	4.62	2.12	2.14	9.88	7.65	5.61
Mean	–	–	7.6	2.7	2.6	2.13	1.36	0.95	9.50	7.39	5.39

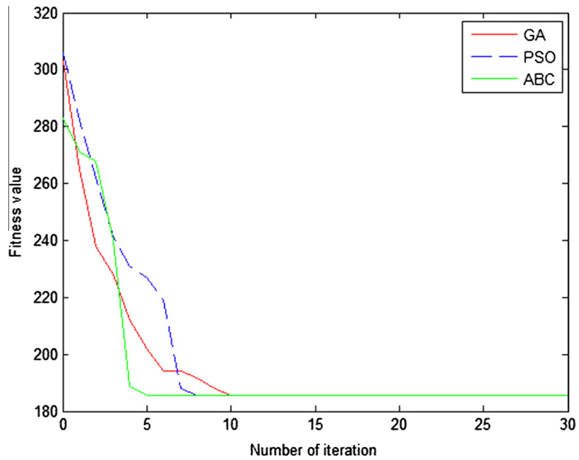


Fig. 9. Convergence curve of genetic algorithm, PSO and ABC algorithms for Example 2.

One of the best options for reducing energy consumption of the nodes and increasing their life time is multi hop. This means, in contrast to the case where sensors send their data directly over long distances towards the base station (Fig. 10), here, they send their data to a neighboring node in the path towards the base station. Therefore, much less energy is wasted. Thus, it is necessary to find the SP between every node and the destination base station.

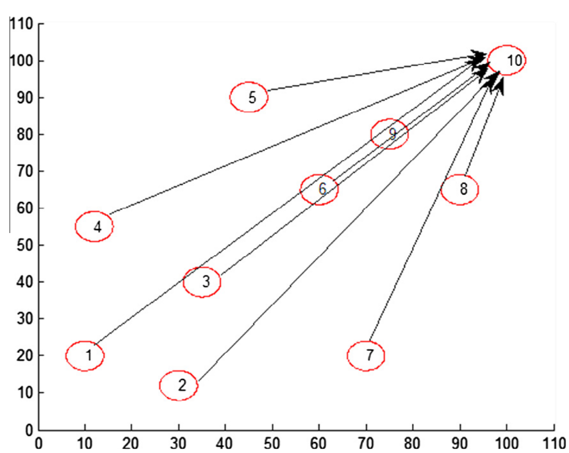


Fig. 10. A wireless sensor network.

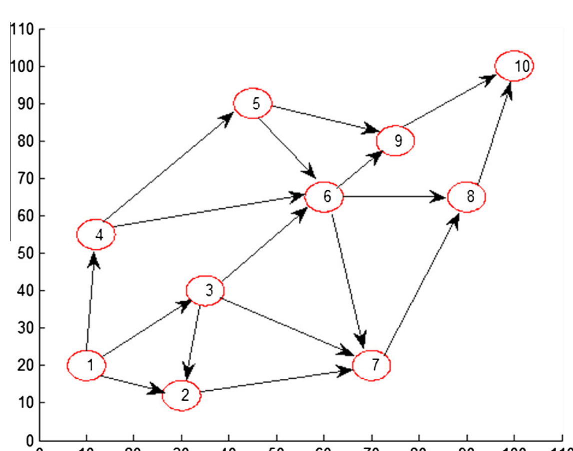


Fig. 11. A WSN with fuzzy lengths.

Table 5
Information corresponding to WSN.

Arcs	Membership function	Arcs	Membership function
(1, 2)	(14, 22)	(5, 6)	(21, 29)
(1, 3)	(24, 32, 35, 36)	(5, 9)	(24, 32, 35, 36)
(1, 4)	(27, 35)	(6, 7)	(38, 46, 49, 50)
(2, 7)	(33, 41, 44, 45)	(6, 8)	(22, 30)
(3, 2)	(20, 28, 31, 32)	(6, 9)	(13, 21, 24, 25)
(3, 6)	(27, 35, 38, 39)	(7, 8)	(41, 49)
(3, 7)	(32, 40)	(8, 10)	(28, 36)
(4, 5)	(40, 48, 51, 52)	(9, 10)	(24, 32, 35, 36)
(4, 6)	(41, 49, 52, 53)		

However, in many cases, the sensor nodes possess mobility. Consequently, the distance between the sensors is not a crisp value. In such a case, fuzzy numbers can be used to represent the distance between the sensors. Fig. 11 shows an example of a WSN with nine sensor nodes and a base station (number 10). Table 5 shows the arc length between the nodes. The SP from sensor node 1 to the base station can be easily achieved by the proposed ABC algorithm. The path 1–3–6–9–10 with fitness of 391.79 is the SP.

6. Other practical applications

There are a number of transportation applications that use SP algorithms. In some applications SPs need to be quickly identified because of the need for repeated recalculation of the SPs (e.g., vehicle routing and scheduling) or the need for an immediate response (e.g., in-vehicle route guidance systems) [12]. A number of heuristic approaches have been proposed in transportation engineering to decrease the computation time of the SP algorithm for vehicle routing and scheduling or in-vehicle route guidance systems. These conventional transportation engineering algorithms assume precise values for the weights of the edges in the transportation network. The approach proposed in this study could be applied to situations where the information about the weights of the edges is imprecise due to incomplete or uncertain information.

Another application for the SP algorithms is traffic-light networks where the goal is to find the SPs in a transportation system with traffic-light controls in a number of road intersections [6]. In this transportation system, the network represents the city, the nodes correspond to the intersections and the arcs correspond to the roads in the city. The goal in this transportation network is to find a SP from one node to another node, where some nodes are constrained to the traffic-light controls. Fuzzy set theory and the approach proposed in this study can be used to solve SP traffic-light network problems where the time to travel through the road fluctuates with traffic conditions.

The third example for the SP algorithms is motion planning for car-like robots in a dynamic environment [33]. In a decentralized system, the goal is to independently determine a SP for each robot to avoid collisions among them. These problems often assume each robot moves with a fixed velocity and each obstacle has a fixed and known geometry. Fuzzy set theory and the approach proposed in this study can be used to solve SP car-like robot problems where robots move with variable velocity.

7. Conclusions and future research directions

In this study, we proposed an ABC as a simple and robust optimization technique for obtaining the FSP weight and the corresponding SP in a network with various types of fuzzy arc weights. The performance of the proposed ABC algorithm has been evaluated with two comparative examples. The results were compared with two existing methods, the GA and the PSO algorithm. We showed the convergence speed of the proposed ABC algorithm is higher than the existing GA and the PSO algorithm. In sum, the results show the ABC algorithm can be successfully applied to solve SP problem with different types of fuzzy arc weights in large problems. Developing a full fuzzy version of the proposed method to the problem of speeding-up the SP in continuous-time dynamic networks [1] would be an interesting stream for future research. FSP problems can emerge from a variety of practical applications. In addition to the application of FSP in WSN problems, we see three other possibilities for future research in traffic-light networks, car-like robots, and in-vehicle route guidance systems.

Acknowledgements

The authors would like to thank the anonymous reviewers and the editor for their insightful comments and suggestions. The first author would also like to thank the office of Vice Chancellor for Research and Technology at Islamic Azad University, Qaemshahr Branch, for their financial support.

References

[1] M.K. Ardakani, M. Tavana, A decremental approach with the A* algorithm for speeding-up the optimization process in dynamic shortest path problems, Measurement 60 (2015) 299–307.
[2] E.M. Bernardino, A.M. Bernardino, J.M. Sánchez-Pérez, J.A. Gómez Pulido, M.A. Vega Rodríguez, Swarm optimisation algorithms applied to large balanced communication networks, J. Network Comput. Appl. 36 (1) (2013) 504–522.
[3] M. Canayaz, A. Karcı, Investigation of cricket behaviours as evolutionary computation for system design optimization problems, Measurement 68 (2015) 225–235.
[4] T.N. Chuang, J.Y. Kung, The fuzzy shortest path length and the corresponding shortest path in a network, Comput. Oper. Res. 32 (6) (2005) 1409–1428.
[5] T.N. Chuang, J.Y. Kung, A new algorithm for the discrete fuzzy shortest path problem in a network, Appl. Math. Comput. 174 (1) (2006) 660–668.
[6] Y.-L. Chen, H.-H. Yang, Shortest paths in traffic-light networks, Transp. Res. Part B 34 (2000) 241–253.
[7] W. Chen, X. Jiang, X. Li, J. Gao, X. Xu, S. Ding, Wireless sensor network nodes correlation method in coal mine tunnel based on Bayesian decision, Measurement 46 (2013) 2335–2340.
[8] Y. Deng, Y. Chen, Y. Zhang, S. Mahadevan, Fuzzy Dijkstra algorithm for shortest path problem under uncertain environment, Appl. Soft Comput. 12 (3) (2012) 1231–1237.
[9] Y. Dou, L. Zhu, H.S. Wang, Solving the fuzzy shortest path problem using multi-criteria decision method based on vague similarity measure, Appl. Soft Comput. 12 (6) (2012) 1621–1631.
[10] D. Dubois, H. Prade, Fuzzy Sets and Systems: Theory and Applications, Academic Press, New York, 1980.
[11] A. Ebrahimnejad, Z. Karimnejad, H. Alrezaamiri, Particle swarm optimization algorithm for solving shortest path problems with mixed fuzzy arc weights, Int. J. Appl. Decision Sci. 8 (2) (2015) 203–222.
[12] L. Fu, D. Sun, L.R. Rilett, Heuristic shortest path algorithms for transportation applications: state of the art, Comput. Oper. Res. 33 (2006) 3324–3343.
[13] J. Funck, C. Gühmann, Comparison of approaches to time-synchronous sampling in wireless sensor networks, Measurement 56 (2014) 203–214.
[14] R. Hassanzadeh, I. Mahdavi, N. Mahdavi-Amiri, A. Tajdin, A genetic algorithm for solving fuzzy shortest path problems with mixed fuzzy arc lengths, Math. Comp. Model. 57 (1–2) (2013) 84–99.
[15] F. Hernandes, M.T. Lamata, J.L. Verdegay, A. Yamakami, The shortest path problem on networks with fuzzy parameters, Fuzzy Sets Syst. 158 (14) (2007) 1561–1570.
[16] G. Haosong, Y. Younghwan, Distributed bottleneck node detection in wireless sensor networks, in: IEEE 10th International Conference on Computer and Information Technology (CIT), July 2010, pp. 218–224.
[17] Q. Huang, B. Tang, L. Deng, J. Wang, A divide-and-compress lossless compression scheme for bearing vibration signals in wireless sensor networks, Measurement 67 (2015) 51–60.
[18] X. Ji, K. Iwamura, Z. Shao, New models for shortest path problem with problem with fuzzy arc lengths, Appl. Math. Model. 31 (2) (2007) 259–269.
[19] D. Karaboga, An idea based on honey bee swarm for numerical optimization Technical Report-TR06, Erciyes University, Engineering Faculty, Computer Engineering Department, 2005.
[20] D. Karaboga, C. Ozturk, A novel clustering approach: artificial bee colony (ABC) algorithm, Appl. Soft Comput. 11 (1) (2011) 652–657.
[21] C.M. Klein, Fuzzy shortest paths, Fuzzy Sets Syst. 39 (1) (1991) 27–41.
[22] A. Kumar, M. Kaur, A new algorithm, for solving shortest path problem on a network with imprecise edge weight, Appl. Appl. Math.: Int. J. 6 (2) (2011) 602–619.
[23] K.C. Lin, M.S. Chern, The fuzzy shortest path problem and its most vital arcs, Fuzzy Sets Syst. 58 (3) (1993) 343–353.
[24] L. Ma, K. Hu, Y. Zhu, H. Chen, Cooperative artificial bee colony algorithm for multi-objective RFID network planning, J. Network Comput. Appl. 42 (2014) 143–162.
[25] I. Mahdavi, R. Nourifar, A. Heidarzade, N. Mahdavi-Amiri, A dynamic programming approach for finding shortest chains in fuzzy network, Appl. Soft Comput. 9 (2) (2009) 503–511.
[26] S. Okada, Fuzzy shortest path problems incorporating interactivity among paths, Fuzzy Sets Syst. 142 (3) (2004) 335–357.
[27] S. Okada, T. Soper, A shortest path problem on a network with fuzzy arc lengths, Fuzzy Sets Syst. 109 (1) (2000) 129–140.
[28] S. Olariu, Q. Xu, Information assurance in wireless sensor networks, in: Proceedings of the 19th IEEE International Parallel and Distributed Processing Symposium, Denver, CO, 2005, http://dx.doi.org/10.1109/IPDPS.2005.257.
[29] R. Özdağ, A. Karcı, Probabilistic dynamic distribution of wireless sensor networks with improved distribution method based on electromagnetism-like algorithm, Measurement 79 (2016) 66–76.
[30] T.S.D. Singh, A. Chatterjee, MMSE design of nonlinear Volterra equalizers using artificial bee colony algorithm, Measurement 46 (2013) 210–219.
[31] A. Tajdin, I. Mahdavi, N. Mahdavi-Amiri, B. Sadeghpour-Gildeh, Computing a fuzzy shortest path in a network with mixed fuzzy lengths using α-cut, Comput. Math. Appl. 60 (2) (2010) 989–1002.
[32] P.M. Wightman, M.A. Labrador, Topology maintenance: extending the lifetime of wireless sensor networks, in: Proc. Latincom, 2009, pp. 1–6.
[33] E.K. Xidias, P.N. Azariadis, N.A. Aspragathos, Time optimum motion planning for a set of car-like robots, IFAC Proc. Volumes 42 (16) (2009) 542–547.
[34] Y. Yu, W. Kang, C. Zhang, J. Wang, J. Ou, A stochastic analysis framework for a steel frame structure using wireless sensor system measurements, Measurement 69 (2015) 202–209.
[35] Y. Zhang, Z. Zhang, Y. Deng, S. Mahadevan, A biologically inspired solution for fuzzy shortest path problems, Appl. Soft Comput. 13 (5) (2013) 2356–2363.