



A novel meta-heuristic based method for deriving priorities from fuzzy pairwise comparison judgments



Ali Mohtashami*

Department of Industrial Management, Qazvin Branch, Islamic Azad University, Nokhbegan Blvd., Qazvin P.O. Box: 34185-1416, Iran

ARTICLE INFO

Article history:

Received 12 June 2013

Received in revised form 5 February 2014

Accepted 28 May 2014

Available online 5 June 2014

Keywords:

Priority vector

Genetic algorithm

Triangular fuzzy number

Trapezoidal fuzzy number

Triangular shaped fuzzy number

Trapezoidal shaped fuzzy number

ABSTRACT

This paper proposes a new method to derive the priority vector from fuzzy pairwise comparison matrices. Unlike several known methods, the proposed method derives crisp weights from consistent and inconsistent fuzzy comparison matrices. Therefore, the crisp weights obviate the need of additional aggregation and ranking procedures. To derive the priority vector, a Modified Fuzzy Logarithmic Least Square Model (MFLLSM) is proposed. In order to solve the MFLLSM, a framework based on genetic algorithm is proposed. In the proposed framework, a heuristic algorithm of population initialization, a heuristic algorithm for simulating fuzzy numbers and a heuristic algorithm of fitness evaluation are proposed.

The solution of the prioritization problem requires finding priorities such that their ratio approximately satisfies the initial judgments. Computational results reveal the superiority of the proposed method in comparison with five well known methods of literature from the viewpoint of satisfaction of initial judgments by the obtained priority vector. It is shown by ten different examples that the deviation of the priorities ratio from initial judgments in the proposed method is less than five existing methods of literature. In addition, unlike several methods of literature, the proposed method considers fuzzy judgments represented by both triangular and trapezoidal fuzzy numbers. Furthermore, the proposed method for the first time considers judgments represented by triangular shaped fuzzy numbers and trapezoidal shaped fuzzy numbers which are discussed in the paper.

© 2014 Elsevier B.V. All rights reserved.

1. Introduction

Analytical Hierarchy Process (AHP) [1] has been extensively used as a Multi Criteria Decision-Making (MCDM) tool or a weight estimation technique in many areas such as selection, evaluation, planning and development, decision making, forecasting, and so on [2].

Triantaphyllou and Lin [3] summarized the following advantages for AHP: (1) AHP is easier to understand and can effectively handle both qualitative and quantitative data; (2) the AHP can also help decision makers to organize the critical aspects of a problem in a hierarchical structure, making the decision process easy to handle; (3) pairwise comparisons in the AHP are often preferred by the decision makers, allowing them to derive weights of criteria and scores of alternatives from comparison matrices rather than quantify weights/scores directly; (4) AHP can be combined with well-known operation research techniques to handle more difficult problems; (5) it can measure the consistency in the

decision maker's judgments. It should be noted that AHP is not the only available method to measure consistency of preferences. Herrera-Viedma et al. [4] proposed a new characterization of the consistency property defined by the additive transitivity property of the fuzzy preference relations. Using this new characterization they proposed a method for constructing consistent fuzzy preference relations from a set of $n - 1$ preference data. They stated that applying this method it is possible to assure better consistency of the fuzzy preference relations provided by the decision makers, and in such a way, to avoid the inconsistent solutions in the decision making processes. Fedrizzi and Giove [5] proposed a new method for calculating the missing elements of an incomplete matrix of pairwise comparison values for a decision problem. The matrix is completed by minimizing a measure of global inconsistency, thus obtaining a matrix which is optimal from the point of view of consistency with respect to the available judgements. Chiclana et al. [6] presented a short discussion on dealing with incomplete information in decision making process and proposed a new estimation method of missing values in an incomplete fuzzy preference relation which is based on the modeling of consistency of preferences via a representable uninorm. Chiclana, et al. [7] analyzed two methods for calculating missing values of an incomplete reciprocal fuzzy

* Tel.: +98 2833665275; fax: +98 2813665279.

E-mail addresses: mohtashami@qiau.ac.ir, mohtashami07@gamil.com

preference relation [5,8]. They stated that the underlying concept driving both methods is the additive consistency property and as a result they claimed that both methods should be seen as complementary rather than competitors in their application. Chiclana et al. [9] stated that consistency of preferences is related to rationality, which is associated with the *transitivity property*. Many properties suggested to model transitivity of preferences are inappropriate for reciprocal preference relations. Therefore, a functional equation was put forward in their paper to model the “cardinal consistency in the strength of preferences” of reciprocal preference relations.

The traditional AHP requires exact judgment. However, due to the complexity and uncertainty involved in real word decision problems, it is sometimes unrealistic or even impossible to require exact judgments. It is therefore more natural or realistic that a decision maker (DM) is allowed to provide fuzzy judgments instead of precise comparisons [10].

However, there are lots of methods dealing with comparison matrices with exact judgments. Grzybowski [11] proposed new optimization techniques for deriving priority weights from comparison matrices. In his proposed approach, the constrained optimization models are based on the same idea which underlies the right eigenvalue method which was proposed by Saaty [12]. Hovanov et al. [13] stated that the problem of deriving weights from pairwise comparison matrices has been treated extensively in the literature, but most of the results are devoted to the case when the matrix under consideration is reciprocally symmetric (i.e., the i, j th element of the matrix is reciprocal to its j, i th element for each i and j). However, there are some applications of the framework when the underlying matrices are not reciprocally symmetric. Therefore, they employed both statistical and axiomatic arguments to derive weights from such matrices. Dong et al. [14] presented two performance measure algorithms to evaluate the numerical scales and the prioritization methods. Using these performance measure algorithms, they compared some common numerical scales and the prioritization methods. Grzybowski [15] proposed a new goal programming model for deriving priority vectors and for measurement of consistency. In his approach the idea of goal programming is combined with the idea of Saaty's eigenvalue method. Bozók et al. [16] studied the extension of the pairwise comparison matrix to the case when only partial information is available: for some pairs no comparison is given and they studied the uniqueness problem of the best completion for two weighting methods, the Eigenvector method and Logarithmic Least Squares method. Huo et al. [17] proposed new parametric methods to determine a family of priority vectors in an analytic hierarchy process, pointing out the logical relation of elements in the comparative matrix. To derive the priorities of alternatives, a series of theorems and mathematical programming models is given based on pairwise comparison matrix. It should be noted that the proposed method of the present paper regards the uncertainty of judgments and for this purpose it considers judgments represented as fuzzy sets.

A number of methods have been developed to deal with fuzzy comparison matrices. The existing methods mainly differ on the employed fuzzy judgments or the developed fuzzy prioritization method [18]. Van Laarhoven and Pedrycz [19] suggested a fuzzy logarithmic least squares method (LLSM) to obtain triangular fuzzy weights from a triangular fuzzy comparison matrix. Boender et al. [20] found that the normalization method in the fuzzy LLSM destroyed the optimality of the fuzzy weights and therefore presented a modification to the normalization method. Kwiesielewicz [21,22] presented a generalized pseudoinverse approach to solve the fuzzy LLSM, which decomposed the fuzzy least squares problem into a crisp modal values problem and an interval analysis problem. Wang et al. [10] revisited the fuzzy logarithmic least square method and pointed out its incorrectness in the normalization of local fuzzy weights, infeasibility in deriving the local fuzzy weights

of a fuzzy comparison matrix when the lower bound value of a non-normalized fuzzy weight turns out to be greater than its upper bound value, uncertainty of local fuzzy weights for incomplete fuzzy comparison matrices, and unreality of global fuzzy weights. Therefore, they suggested a modified Fuzzy LLSM to tackle all these problems. Buckley [23] proposed fuzzy priorities of comparison ratios similar to the proposed approach of Wagenknecht and Hartmann [24], and Buckley employed trapezoidal membership functions, claiming that such numbers are more easily understood by experts. Chang [25] introduced an extent analysis method for the synthetic extent values of the pairwise comparisons and applied a simple arithmetic mean algorithm to find crisp priorities from fuzzy comparison matrices, whose elements are represented by triangular fuzzy numbers. Zhu et al. [26] made a discussion on the extent analysis method and improved the formulation of possibility degree for comparing two triangular fuzzy numbers. The improved formula included the case that two fuzzy numbers do not intersect with each other and defined the degree of possibility of the smaller fuzzy number over the bigger one as zero. These methods have some characteristics.

Firstly, they derive priorities from fuzzy comparison matrices. However, the approach of constructing fuzzy reciprocal matrices, taken by analogy from the crisp prioritization methods leads to some problems, as illustrated for Triangular Fuzzy Numbers (TFNs) in [27] and illustrated for Trapezoidal Fuzzy Numbers (TrFNs) in the next section of the present paper.

Secondly, except for Chang [25] and Zhu et al. [26], all these methods derive fuzzy priorities and, after aggregating, the final scores of the alternatives are also represented as fuzzy numbers or fuzzy sets. Due to the large number of multiplication and addition operations, the resulting fuzzy scores have wide supports and overlap over a large range [18]. As shown by [20,28], the normalization procedure used in some of these methods may even result in irrational final fuzzy scores, where the normalized upper value is less than the normalized mean value, which is less than the normalized lower value.

Thirdly, as it explained in [18], the fuzzy prioritization methods mentioned above require an additional fuzzy ranking procedure in order to compare the final fuzzy scores. The different ranking procedures, however, often give different ranking results [29].

More recently, some methods are proposed which derive priority weights from fuzzy pairwise comparison matrices. Xu and Chen [30] stated that how to derive the interval priority weights from an interval fuzzy preference relation is an interesting and important issue in decision making with interval fuzzy preference relations. They proposed some linear programming models for deriving the priority weights from various interval fuzzy preference relations. Wang et al. [31] based on the definition of additive consistent fuzzy preference relations developed a new approach to obtain priority weights. They proposed linear programming models for generating interval priority weights from additive consistent or inconsistent interval fuzzy preference relations. Genç et al. [32] proposed an approach to discover whether an interval fuzzy preference relation is consistent or not, and to derive interval priority vector of a consistent interval fuzzy preference relation. Lan et al. [33] provided the relations between a multiplicative consistent interval fuzzy preference relation and an additive consistent interval fuzzy preference relation. Based on the new relations, they proposed a new method to derive interval weights by transforming a multiplicative consistent interval fuzzy preference relation into an additive consistent interval fuzzy preference relation, collecting additive consistent information, transforming back into multiplicative consistent information and calculating the interval weights. Xu [34] focuses on the investigation of consistency of interval fuzzy preference relations. He first established a quadratic programming model by minimizing all the deviations of individual interval fuzzy

preference relations and collective interval fuzzy preference relation, from which an exact solution can be found to derive the importance weights of experts. Then, he gave two approaches to constructing additive and multiplicative consistent interval fuzzy preference relations, respectively, and shows the relationship between the consistency of individual interval fuzzy preference relations and the consistency of collective interval fuzzy preference relation. Wang and Chen [35] in order to alleviate inconsistencies applied fuzzy linguistic preference relations to construct a pairwise comparison matrix with additive reciprocal property and consistency. They reviewed the fuzzy AHP method, and then proposed fuzzy linguistic preference relations method. The methods mentioned above obtain fuzzy priorities; therefore, they require an additional fuzzy ranking procedure in order to compare the final fuzzy scores. As explained in [18] the different ranking procedures, however, often give different ranking results, especially for fuzzy priorities with wide overlap over a large range.

To overcome the shortcomings of some of the fuzzy prioritization methods, Mikhailov [27] developed a fuzzy preference programming (FPP) method for deriving crisp priorities from fuzzy comparison judgments that eliminates some of the drawbacks of the existing fuzzy prioritization methods. This approach does not require the construction of complete fuzzy comparison matrices and it can derive crisp priorities from an incomplete set of fuzzy judgments. The proposed approach is also invariant to the specific form of the fuzzy sets used to represent the judgments. By using α -cuts, the initial fuzzy judgments are transformed into a series of interval judgments. The FPP method is employed to transform the prioritization problem into a fuzzy linear programming which can derive priorities from the interval judgments, corresponding to each α -cuts level, thus eliminating the need for an additional fuzzy ranking procedure. An aggregation process of the optimal priorities derived at the different α -level is also needed for obtaining overall crisp scores of the prioritization elements. However, the steps in the FPP method are complicated from a computational point of view [18,27]. Furthermore, determining different α -cuts level and different tolerance parameters affects the final priority vector [27]. In order to avoid some of these steps, Mikhailov proposed a non-linear FPP for prioritization which can find directly crisp values of priorities from a set of comparison judgments, represented as only triangular fuzzy numbers. He stated that in contrast to the linear FPP, which is linear and can be solved using the simplex method, the solution of the non-linear FPP needs some appropriate numerical methods for non-linear optimization to be employed because there is non-linearity in constraints of the model [27]. Moreover, it is shown in [27] that if the fuzzy judgments are skewed or inconsistent, the results of the linear-FPP and non-linear FPP are different.

More recently, Javanbarg et al. [18] proposed a method which derives crisp weights from consistent and inconsistent fuzzy comparison matrices. The Javanbarg et al. [18] method transforms a fuzzy prioritization problem into a constrained non-linear optimization model. They applied an improved particle swarm optimization (PSO) to solve the optimization model. The method of Javanbarg et al. [18] like the non-linear FPP method of [27] finds crisp values of priorities from a set of comparison judgments, represented as only triangular fuzzy number. In other words, the non-linear FPP method of [27] and also method of [18] cannot consider judgments represented by the other types of fuzzy numbers or fuzzy sets except for triangular fuzzy numbers.

This idea – deriving crisp priorities from fuzzy judgment matrices – shows a new way to deal with the prioritization problem from fuzzy reciprocal comparisons in the fuzzy AHP [18].

In order to overcome some of the shortcomings of the prioritization methods above, this paper proposes a prioritization method, which derives exact priorities from consistent and inconsistent fuzzy comparison matrices. Since the proposed method is

based on simulating the vague judgments, it is invariant to the specific form or the shape of the fuzzy sets. In other words, the proposed method can consider judgments represented by symmetric or non-symmetric TFN/TrFN, symmetric or non-symmetric triangular shaped fuzzy number (TSFN)/trapezoidal shaped fuzzy number (TrSFN) and interval value fuzzy set. This property provides opportunity for solving prioritization problems with mixed types of comparison judgments (as it is illustrated in Section 7.3), such as fuzzy sets of different forms and intervals.

It should be noted that the weights role of elements is vital in AHP. In fact, inappropriate weights of elements may results in inaccuracy of ranking of alternatives as it is illustrated in [10]. Therefore, an inappropriate method of prioritization may lead to a wrong decision to be made about the rank of each alternative.

This paper unlike the existing methods illustrates its merit by comparing the obtained priority vector of the proposed method with five well known methods of literature from the viewpoint of satisfaction of initial judgments by the obtained priority vector.

This paper is organized as follows: Section 2 discusses the preference representation formats, Section 3 explains the shortcoming of constructing fuzzy reciprocal matrices in some existing methods, Section 4 provides the proposed fuzzy optimization model, in Section 5 the proposed algorithm of fuzzy number simulating is presented, Section 6 provides the proposed algorithm to obtain priority vector of fuzzy comparison matrices, in Section 7 some numerical examples are presented to illustrate the proposed method and to show the superiority of the proposed method over some of the existing methods, Section 8 concludes the paper.

2. Preference representation formats

To reach a decision, experts have to express their preferences by means of a set of evaluations over a set of alternatives [7]. Different alternatives preference elicitation methods were compared in [36], where it was concluded that pairwise comparison methods are more accurate than nonpairwise methods.

The introduction of the concept of a fuzzy set as an extension of the classical concept of a set when applied to a binary relation leads to the concept of a fuzzy relation. A fuzzy preference value can be associated with the following two semantics [37]: “the intensity of preference (to what extent alternative x_i is preferred to alternative x_j)” and “the uncertainty about the preference (how sure it is that x_i is preferred to x_j)”. The fuzzy interpretation of intensity of preference was introduced by Bezdek et al. [38] via the concept of a reciprocal fuzzy relation and later reinterpreted by Nurmi [39].

Fixed a set of alternatives in a decision making problem, there exist several preference representation formats that can be used by experts to provide their preferences about that set of alternatives. The most common ones are:

- *Preference orderings.* The preferences of an expert about a set of feasible alternatives X are described as a preference ordering $O = \{o(1), \dots, o(n)\}$ where $o(\cdot)$ is a permutation function over the indexes set $\{1, \dots, n\}$ for this expert [40,41]. Thus, an expert gives an ordered vector of alternatives from best to worst.
- *Utility values.* An expert provides his/her preferences about a set of feasible alternatives X by means of a set of n utility values $U = \{u_1, \dots, u_n\}$, $u_i \in [0,1]$, the higher the value for an alternative, the better it satisfies experts' objective [41,42].
- *Preference relations.* In this case, experts' preferences on X are described by means of a $P \subset X \times X$ characterized by a function $\mu_p : X \times X \rightarrow D$ where $\mu_p(x_i, x_j) = p_{ij}$ can be interpreted as the preference degree or intensity of the alternative x_i over x_j expressed in the information representation domain D [41,43]. Different types

of preference relations can be used according to the domain used to evaluate the intensity of the preference:

1. **Fuzzy preference relations** [38,39,41,44,45]: If $D=[0,1]$ every value p_{ij} in the matrix P represents the preference degree or intensity of preference of the alternative x_i over x_j : $p_{ij}=1/2$ indicates indifference between x_i and x_j , $p_{ij}=1$ indicates that x_i is absolutely preferred to x_j , and $p_{ij}>1/2$ indicates that x_i is preferred to x_j . It is usual to assume the additive reciprocity property $p_{ij}+p_{ji}=1 \forall i, j$.
2. **Multiplicative preference relations** [46]: If $D=[1/9,9]$ and then every value p_{ij} in the matrix P represents a ratio of the preference intensity of the alternative x_i to that of x_j , i.e., it is interpreted as x_i is p_{ij} times good as x_j : $p_{ij}=1$ indicates indifference between x_i and x_j , $p_{ij}=9$ indicates that x_i unanimously preferred to x_j , and $p_{ij} \in \{2,3,\dots,8\}$ indicates intermediate evaluations. It is usual to assume the multiplicative reciprocity property $p_{ij} \cdot p_{ji}=1 \forall i, j$ too.
3. **Linguistic preference relations** [47–49]: If $D=S$, where S is a linguistic term set $S=\{s_0,\dots,s_g\}$ with odd cardinality $(g+1)$, $s_{g/2}$ being a natural label (meaning “equally preferred”) and the rest of labels distributed homogeneously around it, then every value p_{ij} in the matrix P represents the linguistic preference degree or linguistic intensity of preference of the alternative x_i over x_j .

The relationship between multiplicative and fuzzy preference relations is analyzed in [41] assuming that in a decision making problem an expert, e_k , provides his preferences on X by means of a multiplicative preference relation, $A^k=[a_{ij}^k]$.

In general, if

$$A' = \{A^k = [a_{ij}^k] | a_{ij}^k \cdot a_{ji}^k = 1, a_{ij}^k \in [1/9, 9], k = 1, \dots, m\}$$

is the set of multiplicative preference relations in Saaty's sense, and

$$P' = \{P^k = [p_{ij}^k] | p_{ij}^k + p_{ji}^k = 1, p_{ij}^k \in [0, 1], k = 1, \dots, m\}$$

is the set of additive fuzzy preference relations, then we are looking for a continuous function

$$F: A' \rightarrow P', |F(A^k) = P^k, \forall k.$$

After some mathematical operations (see [41]) Chiclana et al. reached the following results for the relationship between additive and multiplicative preference relations:

Proposition 1. Suppose that we have a set of alternatives, $X=\{x_1,\dots,x_n\}$, and associated with it a multiplicative preference relation $A^k=[a_{ij}^k]$. Then, the corresponding additive fuzzy preference relation, $P^k=[p_{ij}^k]$, associated with A^k is given as follows:

$$p_{ij}^k = f(a_{ij}^k) = \frac{1}{2}(1 + \log_9 a_{ij}^k).$$

In this paper, it is assumed that the evaluations about the alternatives and elements are provided by means of the multiplicative preference relations and every value p_{ij} in the matrix P is represented by TFN, TrFN, TSFN, TrSFN and intervals.

A triangular fuzzy number \tilde{N} , Fig. 1a, is defined by three numbers $a < b < c$ where the vertex of the triangle is at $x=b$ and its base is the interval $[a, c]$. We write $\tilde{N}=(a, b, c)$ for a triangular fuzzy number. A triangular shaped fuzzy number \tilde{M} is partially defined by $a < b < c$ because its sides are continuous curves, not straight lines. We write $\tilde{M} \approx (a, b, c)$ for a triangular shaped fuzzy number \tilde{M} shown in Fig. 1b–e. A trapezoidal fuzzy number \tilde{T} is defined by four number $e < f < g < h$ where $\tilde{T}(x)=1$ on $[f, g]$ and its base is the interval $[e, h]$ as shown in Fig. 1f. We write $\tilde{T}=(e, f, g, h)$ for a

trapezoidal fuzzy number. A trapezoidal shaped fuzzy number \tilde{U} is partially defined by $e < f < g < h$, because its sides need not be straight lines as shown for instance in Fig. 1g and h. We write $\tilde{U} \approx (e, f, g, h)$ for a trapezoidal shaped fuzzy number as shown in Fig. 1g [50].

The concept of considering judgments represented by TSFN and TrSFN is introduced in this paper for the first time; therefore, it is discussed in the following.

Suppose a DM regards the first element (e_1) as “approximately three” times more important than the second element (e_2), i.e. $\tilde{a}_{12}=\tilde{3}$. All the existing methods consider this preference as a TFN (Fig. 1a) because it is simple to work with TFNs and there is no need of complex computations for considering judgments represented by TFNs, whereas “approximately three” means different things to different people from the viewpoint of emphasis on “three” and the viewpoint of tendency to the right or to the left side of “three”. Therefore, an appropriate prioritization method is needed for considering these types of judgments. To deal with this concept, this paper proposes the use of judgments represented by TSFN/TrSFN.

In general, “approximately three” may be expressed by one of the following:

- (i) TFN (Fig. 1a);
- (ii) TSFN for much emphasis on “three” (Fig. 1b), when the certainty of “three” is high for DM;
- (iii) TSFN for less emphasis on “three” (Fig. 1c), when the certainty of “three” is low for DM;
- (iv) TSFN (Fig. 1d), when DM tends to the right side of “three”;
- (v) TSFN (Fig. 1e), when DM tends to the left side of “three”.

It is seen that the membership function of the fuzzy numbers in cases ii, iii, iv and v is nonlinear and judgments represented as TSFN. In Section 7.3, the superiority of the proposed method to the linear FPP method [27] is illustrated for problems with judgments represented as TSFN.

In case i, the membership function of the fuzzy number is linear and there are lots of methods that can regard it, but in Section 7, it is illustrated that in such problems, the proposed method is superior to the five known existing methods of literature from the viewpoint of satisfaction of initial judgments by the obtained priority vector. In other words, the deviation of w_i/w_j from a_{ij} by the proposed method is less than five known methods of literature.

Suppose that A is a 3×3 matrix obtained from pairwise comparisons in the form

$$A = \begin{bmatrix} 1 & a_{12} & a_{13} \\ 1/a_{12} & 1 & a_{23} \\ 1/a_{13} & 1/a_{23} & 1 \end{bmatrix}.$$

The aim is to find the priority vector $(w_1, w_2, w_3)^T$ and a positive reciprocal matrix X in the form

$$X = \begin{bmatrix} 1 & w_1/w_2 & w_1/w_3 \\ w_2/w_1 & 1 & w_2/w_3 \\ w_3/w_1 & w_3/w_2 & 1 \end{bmatrix},$$

which deviation of $a_{12}, a_{13}, 1/a_{12}, a_{23}, 1/a_{13}$, and $1/a_{23}$ from $w_1/w_2, w_1/w_3, w_2/w_1, w_2/w_3, w_3/w_1$, and w_3/w_2 , respectively, are as less as possible; therefore, an appropriate method of estimating priority vector is one that results in $a_{ij}=w_i/w_j$ for all i and j .

3. Fuzzy comparison matrices

Mikhailov [27] explained that the approach of constructing fuzzy reciprocal matrices with judgments represented by TFNs,

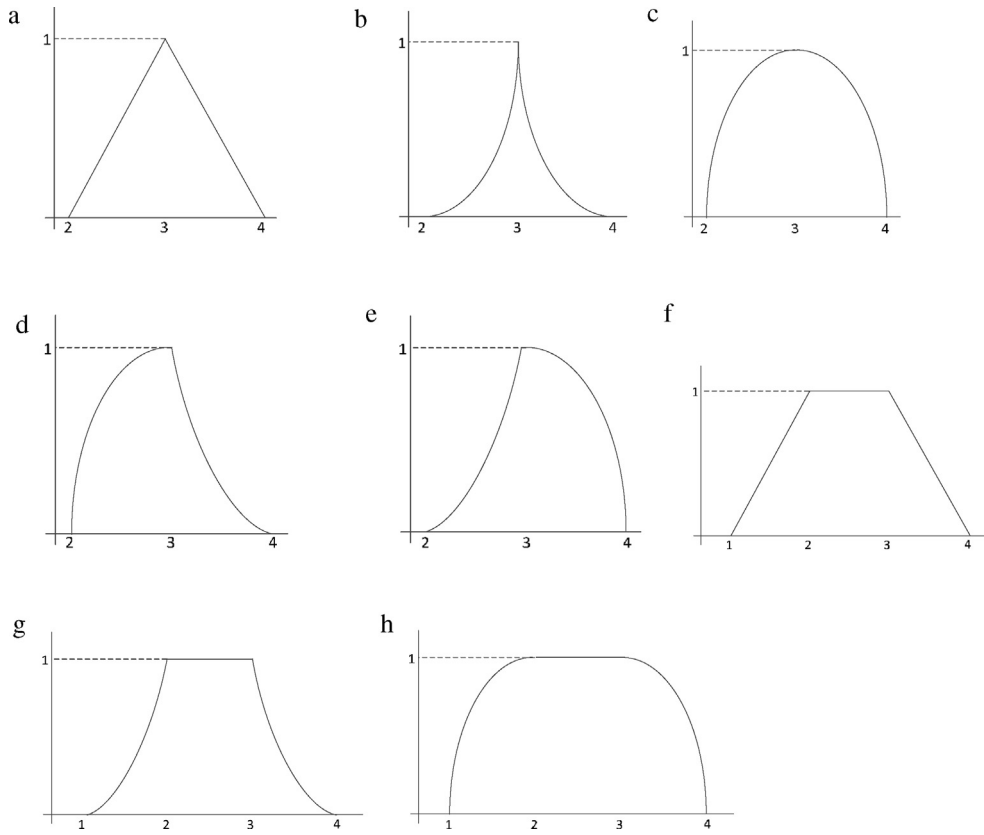


Fig. 1. Different types of a fuzzy judgment.

taken by analogy from the crisp prioritization methods leads to some problems. Now, in this section it is shown that this shortcoming also applies in case of representing judgments by TrFNs.

Let \tilde{N} be a TrFN (e, f, g, h) and has a linear piecewise continuous membership function $\mu_{\tilde{N}}(x)$ with the following characteristics:

- (1) a continuous mapping from \mathbb{R} to the closed interval $[0,1]$;
- (2) $\mu_{\tilde{N}}(x) = 0$ for all $x \in [-\infty, e]$ and for all $x \in [h, +\infty]$;
- (3) strictly linearly increasing on $[e, f]$ and strictly decreasing on $[g, h]$;
- (4) $\mu_{\tilde{N}}(x) = 1$ for $x \in [f, g]$.

Suppose a prioritization problem with n unknown priorities $w = (w_1, w_2, \dots, w_n)^T$, where the pairwise comparison judgments are represented by trapezoidal fuzzy numbers $\tilde{a}_{ij} = (e_{ij}, f_{ij}, g_{ij}, h_{ij})$.

The known fuzzy prioritization methods require a full set of $m = n(n-1)/2$ comparison judgments, in order to construct a positive reciprocal matrix of pairwise comparisons $\tilde{A} = \{\tilde{a}_{ij}\}$ of the type:

$$\tilde{A} = \begin{bmatrix} 1 & \tilde{a}_{12} & \dots & \tilde{a}_{1n} \\ \tilde{a}_{21} & 1 & \dots & \tilde{a}_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{a}_{n1} & \tilde{a}_{n2} & \dots & 1 \end{bmatrix} \quad (1)$$

where $\tilde{a}_{ji} = 1/\tilde{a}_{ij} = (1/h_{ij}, 1/g_{ij}, 1/f_{ij}, 1/e_{ij})$.

The problem mentioned above is explained by two following examples.

Example 1. Consider a perfectly consistent two-dimensional prioritization problem, where there are only two elements to be compared pairwise. The decision maker regards the first element as approximately between “two” and “three” times more

important than the second one. Suppose that this judgment represented by a symmetrical TrFN, $\tilde{a}_{12} = (1, 2, 3, 4)$ (note that the membership degree for values located outside 1 and 4 are assumed equal to zero). The solution of the prioritization problem by existing fuzzy methods requires finding fuzzy priorities $\tilde{w}_1 > 0$ and $\tilde{w}_2 > 0$, such that their ratio approximately satisfies the initial judgment, i.e. $\tilde{r}_{12} = \tilde{w}_1/\tilde{w}_2 \approx \tilde{a}_{12}$. The normalization constraint $\tilde{w}_1 + \tilde{w}_2 \approx \tilde{1}$ must also be satisfied [27].

In order to apply some of the existing fuzzy prioritization methods we have to construct a fuzzy comparison matrix (Eq. (1)), calculating the reciprocal fuzzy judgment, which is $\tilde{a}_{21} = (1/4, 1/3, 1/2, 1)$. It is evident, however, that the reciprocal fuzzy number is non-symmetrical. This asymmetry of the reciprocal judgments in the fuzzy comparison matrix, constructed from symmetrical elements, leads to non-symmetrical final fuzzy scores. For instance, applying the fuzzy geometric mean method [23], the following fuzzy normalized priorities are obtained:

$$\tilde{w}_1 = (0.3333, 0.5798, 0.8697, 1.3333),$$

$$\tilde{w}_2 = (0.1667, 0.2367, 0.3551, 0.6667).$$

The resulting fuzzy ratio, corresponding to these fuzzy priorities can be easily be calculated, by the division operator, defined in [23,51]:

$$\tilde{r}_{12} = \frac{\tilde{w}_1}{\tilde{w}_2} = (0.5000, 1.6330, 3.6742, 8).$$

It is seen that the ratio \tilde{r}_{12} is rather dissimilar to the initial fuzzy judgment $\tilde{a}_{12} = (1, 2, 3, 4)$. Furthermore, the fuzzy ratio is strongly skewed, in contrast to the initial symmetrical judgment.

The resulting fuzzy priorities can be compared, using some of the fuzzy ranking methods. The crisp value of the fuzzy number $\tilde{N} =$

(a, b, c, d) based on Center of Area (COA) method can be expressed by following relation [52]:

$$\text{defuzz}(\tilde{N}) = \frac{\int x\mu(x)dx}{\int \mu(x)dx} = \frac{-ab + cd + 1/3(d-c)^2 - 1/3(b-a)^2}{-a-b+c+d} \quad (2)$$

The COA method gives the following crisp values of the priorities:

$$w_1 = 0.7890, w_2 = 0.3687$$

Normalizing these priorities results in $w_1 = 0.6815, w_2 = 0.3185$ where the resulting crisp ratio is $r = w_1/w_2 = 2.14$, whereas the defuzzified value of $\tilde{a}_{12} = (1, 2, 3, 4)$ by COA method is equal to 2.5. Also, the amount of $\tilde{r}_{12} = \tilde{w}_1/\tilde{w}_2 = (0.5000, 1.6330, 3.6742, 8)$ by COA method is equal to 3.6040 which is located outside of the interval [2,3] in fuzzy number (1, 2, 3, 4).

Solving the same two-dimensional problem with crisp judgments, $a_{12} = 2.5$ and $a_{21} = 0.4$ by any crisp prioritization method will give us perfectly consistent normalized priorities $w_1 = 0.7143, w_2 = 0.2857$, so that the ratio $r = w_1/w_2 = 2.5$ is equal to the initial judgment.

Example 2. Consider a perfectly consistent two-dimensional prioritization problem, where there are only two elements to be compared pairwise. The decision maker regards the first element as approximately between “three” and “four” times more important than the second one. Suppose that this judgment represented by a non-symmetrical TrFN, $\tilde{a}_{12} = (1, 3, 4, 5)$ (note that the membership value for values located outside 1 and 5 are assumed equal to zero). The solution of the prioritization problem by existing fuzzy methods requires finding fuzzy priorities $\tilde{w}_1 > 0$ and $\tilde{w}_2 > 0$, such that their ratio approximately satisfies the initial judgment, i.e. $\tilde{r}_{12} = \tilde{w}_1/\tilde{w}_2 \approx \tilde{a}_{12}$. The normalization constraint $\tilde{w}_1 + \tilde{w}_2 \approx \tilde{1}$ must also be satisfied.

In order to apply some of the existing fuzzy prioritization methods we have to construct a fuzzy comparison matrix (Eq. (1)), calculating the reciprocal fuzzy judgment, which is $\tilde{a}_{21} = (1/5, 1/4, 1/3, 1)$. Applying the fuzzy geometric mean method [23] as a method which considers judgments represented by TrFN, the following fuzzy normalized priorities can be obtained:

$$\tilde{w}_1 = (0.309, 0.672, 0.896, 1.545),$$

$$\tilde{w}_2 = (0.138, 0.194, 0.259, 0.691).$$

The resulting fuzzy ratio, corresponding to these fuzzy priorities can be easily be calculated, by the division operator, defined in [23,51]:

$$\tilde{r}_{12} = \frac{\tilde{w}_1}{\tilde{w}_2} = (0.447, 2.598, 4.619, 11.18).$$

It is clear that the ratio \tilde{r}_{12} is rather dissimilar to the initial fuzzy judgment $\tilde{a}_{12} = (1, 3, 4, 5)$.

The resulting fuzzy priorities can be compared, using some of the fuzzy ranking methods.

The COA method gives the following crisp values of the priorities:

$$w_1 = 0.872, w_2 = 0.345.$$

Normalizing these priorities results in $w_1 = 0.716, w_2 = 0.284$, where the resulting crisp ratio is $r = w_1/w_2 = 2.526$, whereas the defuzzified value of $\tilde{a}_{12} = (1, 3, 4, 5)$ by COA method is equal to 3.2. Moreover, the amount of $\tilde{r}_{12} = \tilde{w}_1/\tilde{w}_2 = (0.447, 2.598, 4.619, 11.18)$ by COA method is equal to 4.962

which is located outside of the interval [3,4] in fuzzy number (1, 3, 4, 5).

According to these two examples, it can be concluded that the introduction of additional reciprocal elements and the construction of full fuzzy comparison matrices with judgments represented by TrFNs lead to some inaccuracy in the final results from the viewpoint of deviation of the priorities ratio from initial judgments. Moreover, Mikhailov [27] stated that the skewed reciprocals might reverse the final ranking of the elements if an inverse ratio scale is used. It is proved by Barzilai [53] that the geometric mean method is independent of scale inversion and preserves rank strongly for crisp comparison matrices. But it is proved by Mikhailov [27] that this property of the geometric mean does not hold in the fuzzy cases, even if the pairwise comparisons are perfectly consistent. Therefore, Mikhailov [27] stated that an appropriate prioritization method is one that avoids construction of full fuzzy comparison matrices. The proposed method of this paper like linear FPP and non-linear FPP proposed by Mikhailov [27] derives the priority vector from a fuzzy comparison matrix without need of constructing full fuzzy comparison matrices, but since the reciprocity property is imperative in the AHP method, the proposed method regards this property by a heuristic method proposed in Section 6. For this purpose, in Section 6 it is illustrated that the proposed method makes the full comparison matrices with reciprocal elements based on the simulated judgments not directly the fuzzy judgments. Since the proposed method works with simulated judgments, it overcomes the shortcomings of the fuzzy reciprocal elements. Moreover, the proposed heuristic simulation method of fuzzy numbers gives the opportunity of considering symmetrical/non-symmetrical fuzzy judgments and also the judgments represented by interval value fuzzy sets.

4. Proposed fuzzy optimization model

The prioritization method refers to the process of deriving a priority vector from the numerical pairwise comparison matrix. Dong et al. [14] stated that there have been a large number of prioritization methods, among which the Eigenvalue method [54] and the logarithmic least square method (LLSM) [55] are most commonly used [14]. Crawford and Williams [55] proposed the LLSM. They stated that the LLSM shares the desirable qualities of the Eigenvalue method and is preferable in several important aspects.

The LLSM method [55] uses the L^2 metric in defining an objective function of the following optimization problem:

$$\begin{cases} \min & \sum_{i=1}^n \sum_{j=1}^n \left[\ln(a_{ij}) - \ln\left(\frac{w_i}{w_j}\right) \right]^2 \\ \text{s.t.} & w_i \geq 0, \quad \sum_{i=1}^n w_i = 1 \end{cases}$$

The proposed fuzzy optimization model of this paper is inspired by the LLSM [55]. Suppose that a fuzzy judgment matrix \tilde{A} is constructed as Eq. (1). The elements of the judgment matrix are pairwise comparisons ratio represented by TFNs, TrFNs, TSFNs, TrSFNs, intervals or a combination of them with extreme points of l_{ij} and u_{ij} .

The proposed fuzzy optimization model to derive crisp priorities from fuzzy judgment matrix is as follows:

$$\min J = \sum_{i=1}^n \sum_{j=1}^n \left(\ln\left(\frac{w_i}{w_j}\right) - \ln(\tilde{a}_{ij}) \right)^2 \quad (3)$$

subject to

$$\sum_{k=1}^n w_k = 1 \quad (4)$$

$$l_{ij} \lesssim \frac{w_i}{w_j} \lesssim u_{ij} \quad (5)$$

$$w_k > 0, \quad k = 1, 2, \dots, n \quad (6)$$

The proposed prioritization model is a constrained nonlinear optimization model. The objective function of the optimization model which is represented in Eq. (3) is inspired by LLSM [55] but with judgments represented by fuzzy numbers or fuzzy sets \tilde{a}_{ij} . Eq. (3) obtains exact weights such that it minimizes the deviation of priorities ratio (w_i/w_j) from initial judgments (\tilde{a}_{ij}). Since value of w_i and w_j are exact numbers, calculating the natural logarithm of w_i/w_j is applicable and the natural logarithm of w_i/w_j is an exact number, but in order to compute the natural logarithm of \tilde{a}_{ij} , since \tilde{a}_{ij} is a fuzzy number, an appropriate method of calculating $\ln(\tilde{a}_{ij})$ should be applied.

In order to calculate the objective function (3), since $\ln(\tilde{a}_{ij})$ should be computed, a proposed heuristic method based on a proposed heuristic method of simulating fuzzy numbers is provided in Section 6. Furthermore, the proposed heuristic method of simulating fuzzy numbers is provided independently in Section 5.

Eq. (4) assures that the obtained priority vector is normalized. Since a framework based on genetic algorithm is utilized to solve the optimization model, a heuristic method in Section 6 is provided to generate random numbers for $(w_1, w_2, \dots, w_n)^T$ such that the sum of w_k be equal to one.

An exact priority vector $(w_1, w_2, \dots, w_n)^T$ derived from \tilde{A} must satisfy the fuzzy inequality (5) where $w_i > 0, w_j > 0, i \neq j$ and the symbol \lesssim means “fuzzy less than or equal to”. Consider a set of $m \leq n(n-1)/2$ fuzzy pairwise comparison judgments which the lower and upper limits are denoted by l_{ij} and u_{ij} respectively. When the fuzzy judgments are consistent, there are many priority vectors, whose elements satisfy the inequalities $l_{ij} \leq w_i/w_j \leq u_{ij}$. If the judgments are inconsistent, there is no priority vector that satisfies all fuzzy judgments simultaneously. However, it is reasonable to try and find a vector that satisfies all judgments “as well as possible”. This implies that a good enough solution vector has to satisfy all fuzzy judgments approximately, or $l_{ij} \lesssim w_i/w_j \lesssim u_{ij}$. In Section 6 a heuristic method is provided to hold this constraint with minimum violation for inconsistent judgments.

The proposed fuzzy optimization model is a constrained nonlinear optimization model and there are nonlinear relations in both objective function and constraints. General optimization algorithms limited to convex regular functions cannot be applied to this optimization problem since the model is nonlinear and also the natural logarithm of a fuzzy number \tilde{a}_{ij} should be calculated. Therefore, an efficient optimization algorithm which can solve difficult optimization problems may be employed. For this purpose, a framework based on genetic algorithm is proposed in Section 6.

5. Proposed fuzzy number simulating

Prior to elaborate the proposed algorithm of deriving priorities from fuzzy pairwise comparison judgments, which is described in Section 6, this section sheds light on the proposed method of fuzzy number simulating, which is used in the proposed algorithm (Section 6.3). It should be noted that the proposed method of fuzzy number simulating can be applied for different types of fuzzy numbers including TFN, TrFN, TSFN, TrSFN, interval value fuzzy sets.

Step 1 Let N^l and N^u be the left and the right extreme points of fuzzy number \tilde{N} , respectively.

Step 2 Classify the close interval $[N^l, N^u]$ into C classes with distance v :

$$\underbrace{[N^l, N^l + v]}_{\text{Class 1}}, \underbrace{[N^l + v, N^l + 2v]}_{\text{Class 2}}, \dots, \underbrace{[N^l + (C-1)v, N^l + C.v]}_{\text{Class C}} \quad (7)$$

Step 3 Calculate the membership degree of extreme points of each class.

$$[\mu_N(N^l), \mu_N(N^l + v)], [\mu_N(N^l + v), \mu_N(N^l + 2v)], \dots, [\mu_N(N^l + (C-1)v), \mu_N(N^l + C.v)] \quad (8)$$

Step 4 Calculate the average of the extreme points of relation (8).

$$\begin{aligned} \eta_1 &= \frac{1}{2}(\mu_N(N^l) + \mu_N(N^l + v)), \eta_2 \\ &= \frac{1}{2}(\mu_N(N^l + v) + \mu_N(N^l + 2v)), \dots, \eta_C \\ &= \frac{1}{2}(\mu_N(N^l + (C-1)v) + \mu_N(N^l + C.v)) \end{aligned} \quad (9)$$

Step 5 Normalize $\eta_1, \eta_2, \dots, \eta_C$.

$$\psi_1 = \frac{\eta_1}{\sum_{i=1}^C \eta_i}, \psi_2 = \frac{\eta_2}{\sum_{i=1}^C \eta_i}, \dots, \psi_C = \frac{\eta_C}{\sum_{i=1}^C \eta_i} \quad (10)$$

where ψ_i stands for the normalized value of η_i .

Step 6 Calculate the cumulative values of ψ_i .

$$\delta_1 = \psi_1, \delta_2 = \psi_1 + \psi_2, \dots, \delta_C = \sum_{i=1}^C \psi_i \quad (11)$$

Step 7 Generate a uniform random number $\theta \in [0, 1]$

- If $\theta \leq \delta_1$, generate a uniform random number between $[N^l, N^l + v]$
- If $\theta > \delta_1$ and $\theta \leq \delta_2$, generate a uniform random number between $[N^l + v, N^l + 2v]$

⋮

- If $\theta > \delta_{C-1}$ and $\theta \leq \delta_C$, generate a uniform random number between $[N^l + (C-1)v, N^l + C.v]$

Step 8 Repeat step 7 numerously in order to simulate the fuzzy number \tilde{N} .

5.1. Numerical example for the proposed fuzzy number simulating method

To illustrate the validity of the proposed fuzzy number simulating method, consider two fuzzy numbers $\tilde{a} = (1, 2, 3)_{TFN}$ and $\tilde{b} = (1, 2, 3)_{LRFNLM}$ with membership function defined in Eqs. (12) and (13), respectively. These two fuzzy numbers are simulated numerously by utilizing the proposed fuzzy number simulating method. The left part of Fig. 2 demonstrates the bar chart of number of generated figures in each class for $\tilde{a} = (1, 2, 3)_{TFN}$ and the right part demonstrates the membership function curve for $\tilde{a} = (1, 2, 3)_{TFN}$. The left part of Fig. 3 demonstrates the bar chart of number of generated figures for $\tilde{b} = (1, 2, 3)_{LRFNLM}$ and the right part demonstrates the membership function curve for $\tilde{b} = (1, 2, 3)_{LRFNLM}$. With comparing the left and the right part of Figs. 2 and 3, it is evident that

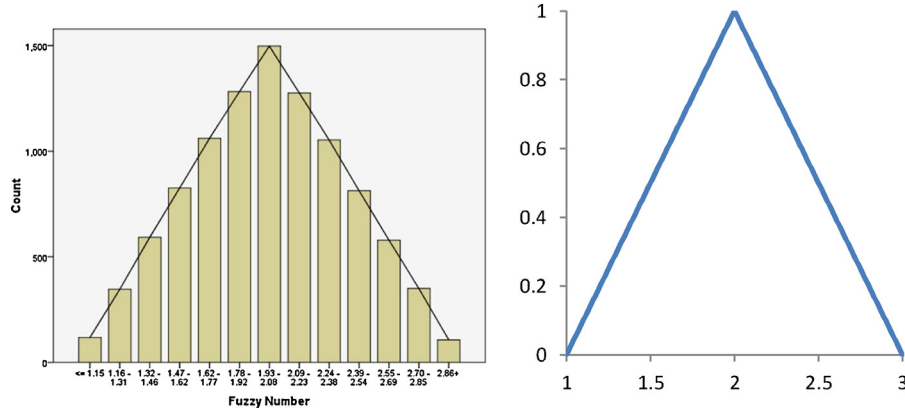


Fig. 2. Bar chart of number of simulated figures in each class (left figure) and the membership function curve (right figure) for $\tilde{a} = (1, 2, 3)_{TFN}$.

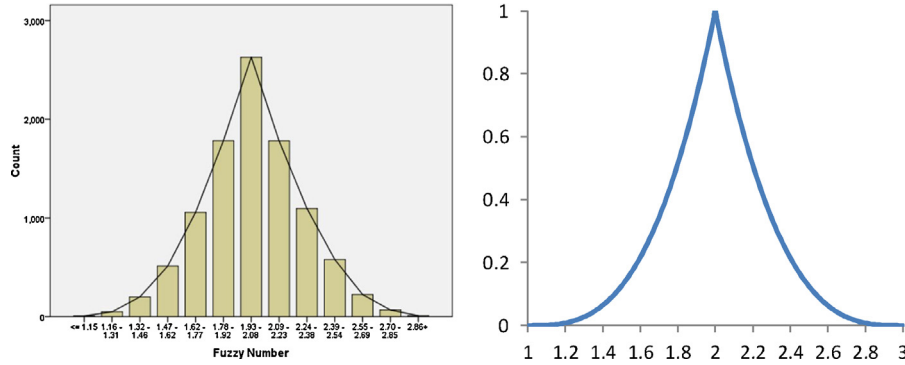


Fig. 3. Bar chart of number of simulated figures in each class (left figure) and the membership function curve (right figure) for $\tilde{b} = (1, 2, 3)_{LRFNLM}$.

the pattern of simulated figures follows the membership function curve; therefore, the validity of the proposed method is accepted.

$$\mu_{\tilde{a}}(x) = \begin{cases} 0 & x < 1 \\ \frac{x-1}{1} & 1 \leq x \leq 2 \\ \frac{3-x}{1} & 2 \leq x \leq 3 \\ 0 & x > 3 \end{cases} \quad (12)$$

$$\mu_{\tilde{b}}(x) = \begin{cases} 0 & x < 1 \\ \left(\frac{x-1}{1}\right)^3 & 1 \leq x \leq 2 \\ \left(\frac{3-x}{1}\right)^3 & 2 \leq x \leq 3 \\ 0 & x > 3 \end{cases} \quad (13)$$

6. The proposed algorithm to derive priority vector

Genetic Algorithms have proven to be very adaptable to a great variety of different complex optimization tasks and many researches use this algorithm to solve the different problems [56–60]. The main role of genetic algorithm (GA) is to perform global search within all feasible search spaces. In the representation of GA, real-number representation is selected instead of bit-string one. Real-number representation has several advantages of (i) being better adapted to numerical optimization problems with continuous decision variables, (ii) speeding up the search over the bit-string representation, and (iii) easing the development of approaches for hybridizing with other conventional methods such as local search techniques [61]. In this section, the pseudo code of

the proposed GA based algorithm is provided to derive priority vector from fuzzy pairwise comparison judgments. Furthermore, the proposed algorithm is explained by Sections 6.1–6.7.

```

L1: "Initialize population"
L2: for c ← 1 to population size
L3:   counter ← 0
L4:   w1c ← a random number ∈ [0, 1]
L5:   do
L6:     counter ← counter + 1
L7:     upper limit ← 0
L8:     for k ← 1 to counter
L9:       upper limit ← wkc + upper limit
L10:    next k
L11:    wk+1c ← a random number ∈ [0, 1 – upper limit]
L12:  loop until (counter ← n – 2)
L13:  upper limit ← 0
L14:  for k ← 1 to counter + 1
L15:    upper limit ← wkc + upper limit
L16:  next k
L17:  wnc ← 1 – upper limit
L18: next c

L19: "Satisfaction of initial judgments"
L20: step ← 0
L21: for c ← 1 to population size
L22:   if (lij ≤ wic/wjc ≤ uij) then go to the next c
L23:   else:
L24:     do
L25:       counter ← 0
L26:       if c ← 1 then
L27:         if (counter1 ← δ) then
L28:           step ← step + ε
L29:           counter1 ← 0
L30:       end if
L31:     end if
L32:     generate a new priority vector
L33:     if (lij – step ≤ wic/wjc ≤ uij + step) then go to the
next c

```



```

L34:         else:
L35:             counter ← 1
L36:             counter1 ← counter1 + 1
L37:         loop until (counter ← 0)
L38:     next c

L39: for t ← 1 to a maximum number of iterations
L40:     "Fitness calculation of parents"
L41:     for f ← 1 to η
L42:         aijf ← a random number using proposed fuzzy number
        simulating
L43:     next f
L44:     for c ← 1 to population size
L45:         sumfitc ← 0
L46:         for f ← 1 to η
L47:             fitnessfc ←  $\sum_{i=1}^n \sum_{j=1}^n ((\ln(w_i^c/w_j^c) - \ln(a_{ijf}))^2 +$ 
 $(\ln(w_j^c/w_i^c) - \ln(1/a_{ijf}))^2)$ 
L48:             sumfitc ← sumfitc + fitnessfc
L49:         next f
L50:         fitnessc ← (1/η) * sumfitc
L51:     next c

L52: "Selection"
L53:     perform selection by tournament strategy
L54: "Crossover"
L55:     perform two-point crossover at rate of crossover rate
    and generate offsprings

L56:     if  $\left(\sum_{k=1}^n w_k^c \neq 1\right)$  then normalize priority vector
L57:     for c ← 1 to population size
L58:         if (lij - step ≤ wic/wjc ≤ uij + step) then go to the next c
L59:         else:
L60:             do
L61:                 counter ← 0
L62:                 generate a new priority vector
L63:                 if (lij - step ≤ wic/wjc ≤ uij + step) then go to the
        next c
L64:             else:
L65:                 counter ← 1
L66:                 loop until (counter ← 0)
L67:             next c

L68: "Mutation"
L69:     for c ← 1 to population size
L70:         swap the amounts of two random genes at rate of
        mutation rate
L71:         if (lij - step ≤ wic/wjc ≤ uij + step) then go to the next c
L72:         else:
L73:             do
L74:                 counter ← 0
L75:                 generate a new priority vector
L76:                 if (lij - step ≤ wic/wjc ≤ uij + step) then go to the
        next c
L77:             else:
L78:                 counter ← 1
L79:                 loop until (counter ← 0)
L80:             next c

L81: "Fitness calculation of offsprings"
L82:     calculate fitness for offsprings and choose the best
        priority vector

L83: "Replacement"
L84:     reproduce the new population by tournament

L85: next t
L86: repeat iterations until a satisfactory solution been found

```

6.1. The proposed initialization

The first section of the proposed algorithm initializes the population (L1–L18). In the proposed heuristic algorithm of population initialization, each chromosome with n genes indicates a priority vector, and each gene indicates the weight of an element of pairwise comparison matrix. The proposed heuristic initialization algorithm

is designed such that the sum of genes' values in each chromosome be equal to one; therefore, constraint (4) be hold. To initialize the population, a random number between [0,1] is inserted into w_1^c (L4), where w_1^c depicts the weight of element 1 in chromosome c . In order to generate w_{k+1}^c (the weight of element $k+1$ in chromosome c), the sum of w_1^c to w_k^c is calculated and called "upper limit" (L8–L10). Afterward, a random number between [0,1- upper limit] is generated and inserted into w_{k+1}^c (L11). By performing this operation, $w_1^c, w_2^c, \dots, w_{n-1}^c$ be generated. In order to generate w_n^c (the weight of element n in chromosome c), the sum of w_1^c to w_{n-1}^c is calculated and inserted into "upper limit" (L14–L16), and then the amount of "1- upper limit" is inserted into w_n^c (L17). By performing this algorithm for each chromosome, $w_1^c, w_2^c, \dots, w_{n-1}^c, w_n^c$ be generated such that the constraint (4) be hold.

6.2. Satisfaction of initial judgments

After population initialization, each chromosome (priority vector) should be evaluated whether it satisfies constraint (5). For this purpose, at first, the proposed algorithm evaluates whether each priority vector satisfies inequalities $l_{ij} \leq w_i^c/w_j^c \leq u_{ij}$ for all judgments (note that \leq is less than or equal to not fuzzy less than or equal to – L22). If the priority vector satisfies these inequalities, it is remained for further analysis. Since there may be different priority vectors that satisfy inequalities $l_{ij} \leq w_i^c/w_j^c \leq u_{ij}$, a more precise evaluation is needed to compare the priority vectors in order to determine which of them better satisfies the initial judgments. This precise evaluation is discussed in Section 6.3, where the fitness of each chromosome is calculated.

If the first chromosome (priority vector) of the population does not satisfy inequalities $l_{ij} \leq w_i^c/w_j^c \leq u_{ij}$ for all judgments, the proposed algorithm replaces it with a new generated chromosome (by the heuristic method of Section 6.1 – L32), and evaluates the replaced chromosome whether it satisfies the inequalities (L33). Again, if the replaced chromosome does not satisfy the inequalities, this chromosome is replaced with a new generated chromosome and evaluation process is repeated. This operation is performed for at most δ times, in order to find a priority vector that satisfies the inequalities for all judgments.

6.2.1. Finding the minimum amount of violation from initial judgments

If the judgments be inconsistent, there is no priority vector that simultaneously satisfies all fuzzy judgments. Therefore, in δ times replacing the first chromosome with a new generated chromosome, if the algorithm does not find a priority vector that satisfies inequalities $l_{ij} \leq w_i^c/w_j^c \leq u_{ij}$ for all judgments, the algorithm lets the priority vector to slightly violates l_{ij} and u_{ij} . For this purpose, a very small amount of ε greater than zero is inserted into "step" (L28), and the algorithm evaluates whether the new generated chromosome (L32) satisfies inequalities $l_{ij}-step \leq w_i^c/w_j^c \leq u_{ij}+step$ for "step = ε " for all judgments (L33). If the first chromosome does not satisfy the inequalities, the proposed algorithm replaces it with a new generated chromosome (by the heuristic method of Section 6.1 – L32), and evaluates the new generated chromosome whether it satisfies the inequalities (L33). Again, if the replaced chromosome does not satisfy the inequalities, this chromosome is replaced with a new generated chromosome and is evaluated. This operation is performed for at most δ times, in order to find a priority vector that satisfies the inequalities.

In δ times replacing the chromosome with a new generated chromosome, if the algorithm does not find a priority vector that satisfies $l_{ij}-step \leq w_i^c/w_j^c \leq u_{ij}+step$ for all judgments for "step = ε ", a small amount of ε greater than zero is added to "step" (L28), and the algorithm tries to find a chromosome that satisfies $l_{ij}-step \leq w_i^c/w_j^c \leq u_{ij}+step$ with the new amount of "step" (note

that now the amount of step is equal to $2^* \varepsilon$). In fact, by adding ε to “step”, the algorithm lets more violation of priority vector from $l_{ij} \leq w_i^c/w_j^c \leq u_{ij}$.

This operation is performed repeatedly until for a given amount of “step”, a priority vector be found that satisfies inequalities $l_{ij} - \text{step} \leq w_i^c/w_j^c \leq u_{ij} + \text{step}$ for all judgments. The more the amount of “step” is, the more the deviation of priorities ratio from initial judgments will be.

With performing this algorithm, the minimum amount of “step” be found that leads to minimize the violation of chromosomes from inequalities $l_{ij} \leq w_i^c/w_j^c \leq u_{ij}$.

Since in this phase, the minimum amount of “step” is determined, and the algorithm finds priority vectors that satisfy inequalities $l_{ij} - \text{step} \leq w_i^c/w_j^c \leq u_{ij} + \text{step}$ for all judgments for a given amount of “step”, this value of “step” is utilized for the other chromosomes and in the remaining of the algorithm (crossover and mutation) whenever it is needed.

Since the process of finding the minimum amount of “step” is performed only once in the entire algorithm, it is suggested to set a large number for δ and a small number for ε at the beginning of the algorithm in order to find as minimum amount for “step” as possible.

It should be mentioned that, at the beginning of evaluating satisfaction of initial judgments, the algorithm enters the process of finding the minimum amount of “step” if and only if the first chromosome does not satisfy inequalities $l_{ij} \leq w_i^c/w_j^c \leq u_{ij}$ for all judgments. In other words, if the first chromosome satisfies inequalities $l_{ij} \leq w_i^c/w_j^c \leq u_{ij}$ for all judgments, the algorithm finds that it can gain priority vectors that satisfy inequalities $l_{ij} \leq w_i^c/w_j^c \leq u_{ij}$ for all judgments; therefore, the algorithm does not enter the process of finding the minimum amount of “step”, and the amount of “step” remains zero for the entire algorithm. To explain more, suppose that the first chromosome satisfies inequalities $l_{ij} \leq w_i^c/w_j^c \leq u_{ij}$ for all judgments but the second chromosome not. In this situation, since the algorithm does not enter the process of finding the minimum amount of “step” and the amount of step is equal to zero, the second chromosome enters into a loop (L24–L37) and it cannot exit the loop until a priority vector be found that satisfy inequalities $l_{ij} \leq w_i^c/w_j^c \leq u_{ij}$ for all judgments. This operation will be performed for all the remained chromosomes.

6.3. The proposed fitness evaluation of chromosomes

The fitness evaluation of chromosomes is performed by means of Eq. (3). To explain the proposed method of fitness evaluation of chromosomes, consider an example of n -dimensional prioritization problem, where there are n elements to be compared pairwise. According to Eq. (3), the following equation should be computed:

$$J = \left(\ln \left(\frac{w_1}{w_2} \right) - \ln(\tilde{a}_{12}) \right)^2 + \left(\ln \left(\frac{w_2}{w_1} \right) - \ln \left(\frac{1}{\tilde{a}_{12}} \right) \right)^2 \\ + \left(\ln \left(\frac{w_1}{w_3} \right) - \ln(\tilde{a}_{13}) \right)^2 + \left(\ln \left(\frac{w_3}{w_1} \right) - \ln \left(\frac{1}{\tilde{a}_{13}} \right) \right)^2 \\ + \left(\ln \left(\frac{w_2}{w_3} \right) - \ln(\tilde{a}_{23}) \right)^2 + \left(\ln \left(\frac{w_3}{w_2} \right) - \ln \left(\frac{1}{\tilde{a}_{23}} \right) \right)^2 \\ + \dots \left(\ln \left(\frac{w_{n-1}}{w_n} \right) - \ln(\tilde{a}_{(n-1)(n)}) \right)^2 \\ + \left(\ln \left(\frac{w_n}{w_{n-1}} \right) - \ln \left(\frac{1}{\tilde{a}_{(n-1)(n)}} \right) \right)^2.$$

In order to compute the above equation, a heuristic method based on simulating fuzzy numbers is proposed. The reasons of proposing this method are explained in Section 6.3.1.

To calculate the fitness of chromosomes by means of Eq. (3), each fuzzy judgment \tilde{a}_{ij} is simulated η times by the step 8 of Section 5, and the simulated values are inserted into a_{ijf} (L41–L43). Afterward, the fitness of each chromosome is calculated η times by Eq. (14) based on η amounts of a_{ijf} (L47). Finally, the average of these η values is considered as fitness of the chromosome (L50).

$$\text{fitness}_f^c = \sum_{i=1}^n \sum_{j=1}^n \left(\left(\ln \left(\frac{w_i^c}{w_j^c} \right) - \ln(a_{ijf}) \right)^2 + \left(\ln \left(\frac{w_j^c}{w_i^c} \right) - \ln \left(\frac{1}{a_{ijf}} \right) \right)^2 \right) \quad (14)$$

6.3.1. The reasons of proposing heuristic fitness evaluation method

There are two reasons for proposing the heuristic fitness evaluation method as follows:

- (i) *Obtaining an exact amount of fitness for chromosomes.* There are some known methods for calculating the natural logarithm of a fuzzy number, e.g. “extension principle”, “alpha-level” and etc., but these methods give a fuzzy number or a fuzzy set for the natural logarithm of a fuzzy number; therefore, by means of the mentioned methods, we will have a fuzzy number or a fuzzy set for the fitness of each chromosome, and then an additional ranking procedure is needed in order to compare the fuzzy fitness of chromosomes. As explained in [18] the different fuzzy ranking procedures, however, often give different ranking results, especially for fuzzy numbers with wide overlap over a large range. On contrary, as it can be seen in the proposed method of fitness evaluation, since the fuzzy judgments are simulated as crisp numbers a_{ijf} , the fitness amount of each chromosome is equal to an exact number (crisp number); hence, the chromosomes can be easily be compared with each other for selection, reproduction and ranking procedures.
- (ii) *Avoiding of constructing fuzzy reciprocal judgments.* As discussed in Section 3, the approach of constructing fuzzy reciprocal judgments $1/\tilde{a}_{ij}$ for calculating $(\ln(w_i/w_j) - \ln(\tilde{a}_{ij}))^2 + (\ln(w_j/w_i) - \ln(1/\tilde{a}_{ij}))^2$, taken by analogy from the crisp prioritization methods leads to some problems. As it can be seen in Eq. (14), since both a_{ijf} and $1/a_{ijf}$ are crisp numbers (the simulated numbers), reciprocal fuzzy judgments are not needed more.

6.3.2. Validating the proposed method of computing the natural logarithm of a fuzzy number by means of the proposed simulation method

Since the heuristic proposed method is based on simulating the fuzzy numbers, the nature of this method is approximate; therefore, for an illustration and justification of the proposed heuristic method for computing the natural logarithm of a fuzzy number, the results of the proposed heuristic method is compared to the Zadeh’s extension principle method [62] for the following five examples:

- (i) The natural logarithm of TFN $\tilde{A}_1 = (6, 7, 8)$ is equal to a triangular fuzzy number $\ln(\tilde{A}_1) = (\ln(6), \ln(7), \ln(8)) = (1.7918, 1.9459, 2.0794)$ by means of the extension principle method, and it is equal to a crisp 1.9443 by means of the heuristic proposed method. Note that to calculate the natural logarithm of $\tilde{A}_1 = (6, 7, 8)$ by means of the heuristic proposed method, \tilde{A}_1 is simulated numerously and the natural logarithm

Before mutation	0.18	0.25	0.12	0.35	0.1	=1
After mutation	0.18	0.35	0.12	0.25	0.1	=1

Fig. 4. The inversion method for mutation.

of the simulated numbers is calculated and then the average of these numbers is considered as the natural logarithm of \tilde{A}_1 .

- (ii) The natural logarithm of TFN $\tilde{A}_2 = (3, 6, 8)$ is equal to a triangular shaped fuzzy number $\ln(\tilde{A}_2) \approx (\ln(3), \ln(6), \ln(8)) \approx (1.0986, 1.7918, 2.0794)$ by means of the extension principle method, and it is equal to a crisp 1.7138 by means of the heuristic proposed method.
- (iii) The natural logarithm of TFN $\tilde{A}_3 = (4, 5, 7)$ is equal to a triangular shaped fuzzy number $\ln(\tilde{A}_3) \approx (\ln(4), \ln(5), \ln(7)) \approx (1.3863, 1.6094, 1.9459)$ by means of the extension principle method, and it is equal to a crisp 1.6672 by means of the heuristic proposed method.
- (iv) The natural logarithm of a TrFN $\tilde{A}_4 = (2, (5/2), (7/2), 4)$ is equal to a trapezoidal fuzzy number $\ln(\tilde{A}_4) = (\ln(2), \ln(5/2), \ln(7/2), \ln(4)) = (0.6931, 0.9163, 1.2528, 1.3863)$ by means of the extension principle method, and it is equal to a crisp 1.0868 by means of the heuristic proposed method.
- (v) The natural logarithm of a TrFN $\tilde{A}_5 = (1, (3/2), (5/2), 4)$ is equal to a trapezoidal shaped fuzzy number $\ln(\tilde{A}_5) \approx (\ln(1), \ln(3/2), \ln(5/2), \ln(4)) \approx (0, 0.4055, 0.9163, 1.3863)$ by means of the extension principle method, and it is equal to 0.7851 by means of the heuristic proposed method.

As it can be seen in the above examples, the solutions of the proposed method lie in the corresponding interval of the solutions of the extension principle method, so the results of the proposed method are reasonable. The extension principle method gives a fuzzy number for the natural logarithm of a fuzzy number while the heuristic proposed method gives a crisp amount for the natural logarithm of a fuzzy number.

As mentioned in Section 6.3.1, since we need a crisp amount for the fitness of each chromosome, the proposed heuristic method is an appropriate method for this purpose. Note that if we want to defuzzify the results of the extension principle method to have the crisp amounts, there are lots of methods to defuzzify triangular fuzzy numbers, triangular shaped fuzzy numbers, trapezoidal fuzzy numbers, trapezoidal shaped fuzzy numbers and intervals which often give different crisp numbers, but by means of the proposed method we obtain only a crisp amount for both natural logarithm of a fuzzy number and fitness of each chromosome; therefore, in this way, we can easily compare the different chromosomes in procedures of selection, reproduction and ranking.

6.4. Selection

GAs use a selection technique to select individuals from the population to insert individual into mating pool. Individuals from the mating pool are used to generate new offspring, with the resulting offspring forming the basis of the next generation. Goldberg and Deb [63] have shown that the tournament selection has better or equivalent convergence and computational time complexity properties when compared to any other reproduction operator that exists in the literature. So, in this study tournament selection operator is used.

6.5. Crossover

At each iteration, pairs of individuals are selected and are recombined into new solutions [64]. Crossover is used to recombine genetic material in two parent chromosomes to produce two child chromosomes that share characteristics of their parents. In this work two-point crossover is applied at rate of probability of crossover.

Since performing crossover may lead the child chromosomes to violate from constraint (4); therefore, at this circumstance the algorithm normalizes the child chromosomes that violate from constraint (4) (L56). Afterwards, the algorithm evaluates whether each child chromosome satisfies inequalities $l_{ij} - \text{step} \leq w_i^c / w_j^c \leq u_{ij} + \text{step}$ for all judgments (note that the minimum amount of “step” is determined in Section 6.2.1). If a chromosome does not satisfy the inequalities, the algorithm seeks a new chromosome to replace by the heuristic method of Section 6.2.1 (L57–L67).

6.6. Mutation

Mutation operator prevents the solution from being excessively localized and as a result increases the possibility of obtaining the global solution [65]. In this algorithm, inversion method is used for mutation. In inversion method, the substring between two randomly selected positions in a chromosome is inverted (Fig. 4).

With applying the inversion method, the mutated chromosome certainly is normalized and does not violate constraint (4) as it is demonstrated in Fig. 4, but the algorithm should evaluate whether each mutated chromosome satisfies inequalities $l_{ij} - \text{step} \leq w_i^c / w_j^c \leq u_{ij} + \text{step}$ for all judgments for predetermined amount of “step”. If a chromosome does not satisfy the inequalities, the algorithm seeks a chromosome to replace by the heuristic method of Section 6.2.1 (L69–L80).

6.7. Replacement and stopping criterion

After genetic operators are applied to generate the offspring, in this paper a replacement based on tournament is used to combining the current generation and offspring for reproduction.

The same stopping criterion as IGA in [66] is used in the proposed method. A maximum number is chosen to be the upper bound of evolving generations.

7. Illustrative examples

In order to illustrate the performance of the proposed method, several cases are provided to compare the proposed method with some existing methods of literature. Examples 3 and 4 of Section 7.1 show the reasonability of the proposed algorithm. In Example 5 of Section 7.1, four cases are solved with judgments represented as TFN to show the superiority of the proposed method to five known methods of literature. In Section 7.2, two examples are solved to show the capability of the proposed method to consider trapezoidal fuzzy judgments, and it is illustrated that the proposed method is superior to two known methods of literature (which are capable of considering judgments represented by TrFN). In Section

Table 1
Priority vectors for Example 3.

	Non-linear FPP [27]	Linear FPP [27]	Buckley [23]	Wang et al. [10]	Proposed method
w_1	0.6666	0.6666	(0.367,0.667,1.098)	(0.5000,0.6666,0.7500)	0.6664
w_2	0.3333	0.3333	(0.210,0.333,0.634)	(0.2499,0.3333,0.5000)	0.3336

Table 2
Priority vectors for Example 4.

	Non-linear FPP [27]	Linear FPP [27]	Buckley [23]	Wang et al. [10]	Proposed method
w_1	0.1093	0.1094	(0.0851,0.1094,0.1459)	(0.1031,0.1094,0.1192)	0.1099
w_2	0.312	0.3126	(0.2349,0.3089,0.4171)	(0.2698,0.3089,0.3601)	0.3125
w_3	0.5785	0.5778	(0.4268,0.5815,0.7756)	(0.5206,0.5815,0.6269)	0.5775

7.3, the capability of the proposed method to consider judgments represented as TSFN is illustrated. Furthermore, four cases are solved with judgments represented as TSFN to show the superiority of the proposed method to a known method of literature.

7.1. Judgments represented by triangular fuzzy numbers

Example 3. To validate the proposed method of this paper, consider a perfectly consistent two-dimensional prioritization problem, where there are only two elements to be compared pairwise. The decision maker regards the first element more important than the second one and his judgment is represented by triangular fuzzy number $\tilde{a}_{12} = (1, 2, 3)$. Table 1 depicts the priority vectors obtained by the proposed method and four existing methods. It is seen that the difference of the priority vector obtained by the proposed method and linear FPP [27]/non-linear FPP [27] is inconsiderable and also the priority vector obtained by the proposed method is located in range of priority vectors obtained by Wang et al. [10] and Buckley [23]; therefore, it can be concluded that the results of the proposed method are reasonable.

Example 4. As an another example to validate the proposed method, consider example that given by Boender et al. [20], where a decision maker provides the matrix of pairwise comparisons for three different criteria by TFNs; $\tilde{a}_{21} = (2.5, 3, 3.5)$, $\tilde{a}_{31} = (4, 5, 6)$ and $\tilde{a}_{32} = (1.5, 2, 2.5)$. Table 2 depicts the priority vectors obtained by the proposed method and four existing methods. It is seen that the difference of the priority vector obtained by the proposed method and linear FPP [27]/non-linear FPP [27] is inconsiderable and also the priority vector obtained by the proposed method is located in the range of priority vectors obtained by Wang et al. [10] and Buckley [23]; therefore, it can be concluded that the results of the proposed method are reasonable.

Example 5. In this example, four cases are evaluated in which the judgments represented by TFNs and some of the judgments are inconsistent. It is shown that the deviation of priorities ratio from initial judgments in the proposed method is less than five existing well-known methods of literature.

Case 1 of Example 5. Consider a decision maker provides the matrix of pairwise comparisons for three different criteria by TFNs; $\tilde{a}_{13} = (4, 5, 6)$, $\tilde{a}_{21} = (2, 3, 4)$, $\tilde{a}_{23} = (6, 7, 8)$. Table 3 depicts the priority vectors obtained by the proposed method and five existing methods of literature.

As discussed in Section 2 and in [55], considering a pairwise comparison matrix $A = [a_{ij}]$, the aim is to find the priority vector $(w_1, w_2, \dots, w_n)^T$ and a positive reciprocal matrix $X = [w_i/w_j]$ which deviation of a_{ij} from w_i/w_j be as less as possible (see Section 2). Since this concept is basis of the AHP, LSM and LLSM are designed based on this concept. Therefore, to compare the proposed method with the existing methods, the deviation of priorities ratios (w_i/w_j) from initial judgments (a_{ij}) is calculated by sum of squared error (SSE) as defined in Eq. (15) where a_{ij} stands for defuzzified amount of \tilde{a}_{ij} by COA method. The last row of Table 3 (SSE) depicts the amount of Eq. (15) for each method. It should be mentioned that in order to calculate Eq. (15) for Buckley [23] and Wang et al. [10] which derive fuzzy weights from comparison matrices, priority vectors are defuzzified by COA method.

$$SSE = \sum_{i=1}^n \sum_{j=1}^n \left(a_{ij} - \frac{w_i}{w_j} \right)^2 \quad (i \neq j) \quad (15)$$

It is clear that the amount of Eq. (15) for the proposed method (2.8610) is less than five other existing methods; therefore, the proposed method is superior to the five other existing methods. Although, there is small difference between the amount of Eq. (15) for the proposed method and the non-linear FPP [27], but it should be mentioned that the non-linear FPP [27] is applicable for only judgments represented by TFNs, but the proposed method is applicable for TFNs, TrFNs, TSFNs, TrSFNs, intervals and a combination of them as it discussed further. Since the weight of the third criterion is equal to zero by the Chang [25], the amount of Eq. (15) is equal to infinity for this method. It means that the method developed by Chang [25] is much weak to satisfy the initial judgments. Note that $w_1/w_3 = \text{infinity}$, and this is dissimilar to $\tilde{a}_{13} = (4, 5, 6)$. Furthermore, $w_2/w_3 = \text{infinity}$, whereas $\tilde{a}_{23} = (6, 7, 8)$.

Table 3
Priority vectors for Case 1 of Example 5.

	Non-linear FPP [27]	Linear FPP [27]	Buckley [23]	Wang et al. [10]	Chang [25]	Proposed method
w_1	0.3076	0.2738	(0.201,0.278,0.404)	(0.225,0.278,0.355)	0.1826	0.3104
w_2	0.6153	0.6492	(0.461,0.649,0.890)	(0.571,0.649,0.702)	0.8173	0.6117
w_3	0.0769	0.0769	(0.055,0.071,0.097)	(0.071,0.071,0.073)	0	0.0777
SSE	3.0313	4.5624	5.3273	5.1173	Infinity	2.8610

SSE: sum of squared error.

Table 4
Priority vectors for Case 2 of Example 5.

	Non-linear FPP [27]	Linear FPP [27]	Buckley [23]	Wang et al. [10]	Chang [25]	Proposed method
w_1	0.3369	0.1989	(0.196,0.265,0.360)	(0.254,0.265,0.275)	0.3536	0.3191
w_2	0.0490	0.1212	(0.042,0.058,0.092)	(0.050,0.058,0.076)	0	0.0703
w_3	0.5409	0.6122	(0.434,0.592,0.798)	(0.564,0.592,0.610)	0.6463	0.5374
w_4	0.0731	0.0675	(0.055,0.083,0.121)	(0.065,0.083,0.100)	0	0.0729
SSE	52.5607	73.09981	43.058	43.357	Infinity	37.206

SSE: sum of squared error.

Table 5
Priority vectors for Case 3 of Example 5.

	Non-linear FPP [27]	Linear FPP [27]	Buckley [23]	Wang et al. [10]	Chang [25]	Proposed method
w_1	0.3103	0.2800	(0.210,0.285,0.406)	(0.232,0.285,0.360)	0.1845	0.3233
w_2	0.0689	0.0666	(0.049,0.062,0.081)	(0.062,0.062,0.062)	0	0.0666
w_3	0.6206	0.6533	(0.470,0.652,0.883)	(0.576,0.652,0.704)	0.8154	0.6099
SSE	4.2847	6.9577	8.1046	7.7265	Infinity	3.9229

SSE: sum of squared error.

To indicate calculation of Eq. (15), consider the following calculation for the proposed method as instance:

$$\begin{aligned}
 E &= \left(a_{12} - \frac{w_1}{w_2}\right)^2 + \left(a_{13} - \frac{w_1}{w_3}\right)^2 + \left(a_{21} - \frac{w_2}{w_1}\right)^2 + \left(a_{23} - \frac{w_2}{w_3}\right)^2 \\
 &\quad + \left(a_{31} - \frac{w_3}{w_1}\right)^2 + \left(a_{32} - \frac{w_3}{w_2}\right)^2 \\
 &= \left(\frac{1}{3} - \frac{0.3104}{0.6117}\right)^2 + \left(5 - \frac{0.3104}{0.0777}\right)^2 + \left(3 - \frac{0.6117}{0.3104}\right)^2 \\
 &\quad + \left(7 - \frac{0.6117}{0.0777}\right)^2 + \left(\frac{1}{5} - \frac{0.0777}{0.3104}\right)^2 + \left(\frac{1}{7} - \frac{0.0777}{0.6117}\right)^2 \\
 &= 2.861
 \end{aligned}$$

Case 2 of Example 5. Consider a decision maker provides the matrix of pairwise comparisons for four different criteria by TFNs; $\tilde{a}_{12} = (3, 4, 5)$, $\tilde{a}_{14} = (7, 8, 9)$, $\tilde{a}_{31} = (4, 5, 6)$, $\tilde{a}_{32} = (7, 8, 9)$, $\tilde{a}_{34} = (3, 4, 5)$, $\tilde{a}_{42} = (1, 2, 3)$. Table 4 depicts the priority vector obtained by the proposed method and five existing methods. The last row of Table 4 (SSE) depicts the amount of Eq. (15) for each method. The amount of SSE for the proposed method is equal to 37.206 which is less than five other existing methods; therefore, the proposed method is superior to the five other methods. In other words, the proposed method better satisfies the initial judgments compared with the five other existing methods.

Case 3 of Example 5. In order to compare the performance of the proposed method to other methods, suppose that a decision maker provides the matrix of pairwise comparisons for three different criteria by TFNs; $\tilde{a}_{12} = (5, 6, 7)$, $\tilde{a}_{31} = (2, 3, 4)$, $\tilde{a}_{32} = (7, 8, 9)$. Table 5 provides the priority vectors obtained by the proposed method and the five existing methods. It is seen that the amount of SSE for the proposed method is equal to 3.9229 which is less

than five other existing methods; therefore, the proposed method is superior to the five other methods.

Case 4 of Example 5. As an another example to illustrate the superiority of the proposed method to some other existing methods, suppose a decision maker compares four elements and provides the following judgments represented by TFNs; $\tilde{a}_{12} = (7, 8, 9)$, $\tilde{a}_{13} = (3, 4, 5)$, $\tilde{a}_{14} = (6, 7, 8)$, $\tilde{a}_{32} = (4, 5, 6)$, $\tilde{a}_{34} = (7, 8, 9)$, $\tilde{a}_{42} = (3, 4, 5)$. Table 6 depicts the priority vector obtained by the proposed method and the five existing methods. The amount of SSE for the proposed method is equal to 90.574 which is less than five other existing methods; therefore, the proposed method is superior to the five other methods.

7.2. Judgments represented by trapezoidal fuzzy numbers

As mentioned before, the proposed method of this paper has got the ability of deriving exact priorities from fuzzy comparison matrices with fuzzy judgments represented as TrFNs as well as TFNs. To illustrate the performance of the proposed method, consider the Example 6.

Example 6. In this example, two cases are evaluated in which the judgments represented by TrFNs. As discussed before, all the existing methods introduced in Section 1 except for Buckley [23] and linear FPP [27] are developed to regard judgments represented by TFNs and do not consider the judgments represented by TrFNs. Therefore, the obtained priority vector by the proposed method for the following examples is compared with the obtained vectors by Buckley [23] and linear FPP [27]. It is shown that the deviation of priorities ratio from initial judgments in the proposed method is less than both linear FPP [27] and Buckley [23].

Case 1 of Example 6. Consider a decision maker provides the matrix of pairwise comparisons for four different elements by TrFNs; $\tilde{a}_{12} = (4, (9/2), (11/2), 6)$; $\tilde{a}_{14} = (7, (15/2), (17/2), 9)$; $\tilde{a}_{31} = (3, (7/2), (9/2), 5)$; $\tilde{a}_{32} = (7, (15/2), (17/2), 9)$; $\tilde{a}_{34} =$

Table 6
Priority vectors for Case 4 of Example 5.

	Non-linear FPP [27]	Linear FPP [27]	Buckley [23]	Wang et al. [10]	Chang [25]	Proposed method
w_1	0.5583	0.6397	(0.456,0.600,0.779)	(0.554,0.600,0.638)	0.6955	0.5262
w_2	0.0519	0.0805	(0.033,0.043,0.059)	(0.039,0.043,0.049)	0	0.0487
w_3	0.3311	0.2221	(0.209,0.275,0.368)	(0.248,0.275,0.309)	0.3044	0.3568
w_4	0.0585	0.0574	(0.061,0.080,0.105)	(0.074,0.080,0.086)	0	0.0680
SSE	121.36	169.76	112.25	112.28	Infinity	90.574

SSE: sum of squared error.

Table 7
Priority vectors for Case 1 of Example 6.

	Linear FPP [27]	Buckley [23]	Proposed method
w_1	0.2423	(0.2097, 0.2404, 0.3177, 0.3686)	0.3249
w_2	0.0721	(0.0336, 0.0381, 0.0504, 0.05912)	0.0511
w_3	0.6274	(0.4568, 0.5251, 0.6841, 0.7796)	0.5589
w_4	0.0579	(0.0616, 0.0703, 0.0916, 0.1051)	0.0649
SSE	45.614	60.6925	35.080

SSE: sum of squared error.

Table 8
Priority vectors for Case 2 of Example 6.

	Linear FPP [27]	Buckley [23]	Proposed method
w_1	0.655362756	(0.4085, 0.4823, 0.6551, 0.7613)	0.555546381
w_2	0.084289265	(0.0412, 0.0475, 0.0660, 0.0803)	0.078396655
w_3	0.187529916	(0.2297, 0.2596, 0.3344, 0.3833)	0.292293601
w_4	0.072818084	(0.0663, 0.0759, 0.0999, 0.1156)	0.073763363
SSE	96.1872	94.5466	71.0065

SSE: sum of squared error.

(6, (13/2), (15/2), 8); $\tilde{a}_{42} = (3, (7/2), (9/2), 5)$. Table 7 illustrates the priority vector obtained by the proposed method and methods of Buckley [23] and linear FPP [27]. The last row of Table 7 (SSE) illustrates the amount of Eq. (15) for each method. The amount of SSE for the proposed method is equal to 35.080 which is less than two other existing methods; therefore, the proposed method is superior to the two other existing methods. In other words, the proposed method better satisfies the initial judgments compared with Buckley [23] and linear FPP [27].

Case 2 of Example 6. In order to compare the performance of the proposed method with other existing methods to deal with judgments represented by TrFNs, suppose that a decision maker provides the matrix of pairwise comparisons for four different criteria by TrFNs; $\tilde{a}_{12} = (2, (5/2), (7/2), 4)$; $\tilde{a}_{13} = (5, (11/2), (13/2), 7)$; $\tilde{a}_{14} = (6, (13/2), (15/2), 8)$; $\tilde{a}_{32} = (6, (13/2), (15/2), 8)$; $\tilde{a}_{34} = (7, (15/2), (17/2), 9)$; $\tilde{a}_{42} = (3, (7/2), (9/2), 5)$. Table 8 provides the priority vector obtained by the proposed method and methods of Buckley [23] and linear FPP [27]. It is seen that the amount of SSE for the proposed method is equal to 71.0065 which is less than two other existing methods; therefore, the proposed method is superior to the two other existing methods.

7.3. Judgments represented by triangular shaped fuzzy numbers/trapezoidal shaped fuzzy numbers

As mentioned before, in some cases, some preferences may be represented by TSFN/TrSFN. In this section, the capability of the proposed method to consider such problems is discussed.

Eq. (16) with δ and $\gamma = 1$ depicts the membership function of a TFN: $\tilde{A} = (a, b, c)$ which the form of the membership function is similar to Fig. 1a. Determining other unequal to one amounts for δ and γ , results in converting a TFN into a TSFN. For instance, Fig. 1b demonstrates a TSFN with δ and $\gamma > 1$; Fig. 1c demonstrates a TSFN with δ and $\gamma < 1$; Fig. 1d demonstrates a TSFN with $\delta > 1$ and $\gamma < 1$ and Fig. 1e demonstrates a TSFN with $\delta < 1$ and $\gamma > 1$. Therefore, by

Table 9
The priority vectors for different amounts of δ and γ .

	$\delta = 1, \gamma = 1$	$\delta = 3, \gamma = 1/3$	$\delta = 1/3, \gamma = 3$
w_1	0.6664	0.6935	0.6324
w_2	0.3336	0.3065	0.3676

Table 10
Priority vectors for Case 1 of Example 8.

	Linear FPP [27]	Proposed method
w_1	0.200239615	0.334098395
w_2	0.11935968	0.061149726
w_3	0.613537382	0.531515509
w_4	0.066863296	0.07323637
SSE	146.7131	83.1927

SSE: sum of squared error.

determining the value of δ and γ unequal to one, DM can express his judgment as a TSFN.

$$\mu_{\tilde{A}}(x) = \begin{cases} 0 & x \leq a \\ \left(\frac{x-a}{b-a}\right)^{\delta} & a \leq x \leq b \\ \left(\frac{c-x}{c-b}\right)^{\gamma} & b \leq x \leq c \\ 0 & x \geq c \end{cases} \quad (16)$$

Example 7. In order to illustrate the capability of the proposed method to consider such problems, consider a perfectly consistent two-dimensional prioritization problem, where there are only two elements to be compared pairwise. Suppose a DM regards the first element as approximately two times more important than the second one $\tilde{a}_{12} = (1, 2, 3)$, and his judgment is represented by Eq. (16). Table 9 indicates the priority vector obtained by the proposed method for different amounts of δ and γ . It is seen that for $\delta = 1$ and $\gamma = 1$, the priority vector is equal to $w_1 = 0.6664$ and $w_2 = 0.3336$, but for $\delta = 3$ and $\gamma = 1/3$, the weight of element 1 is equal to $w_1 = 0.6935$ which is greater than 0.6664 and the weight of element 2 is equal to $w_2 = 0.3065$ which is less than 0.3336 as expected. Also, for $\delta = 1/3$ and $\gamma = 3$, the weight of element 1 is equal to $w_1 = 0.6324$ which is less than 0.6664 and the weight of element 2 is equal to $w_2 = 0.3676$ which is greater than 0.3336 as expected.

Example 8. In this example, four cases are evaluated in which some of the judgments are represented by TSFN. Since linear FPP [27] is capable of considering judgments represented by TSFN/TrSFN, the priority vectors obtained for the four cases by the proposed method is compared with the linear FPP [27]. It is shown by different examples that the deviation of priorities ratio from initial judgments in the proposed method is less than linear FPP [27].

Case 1 of Example 8. Suppose that a decision maker provides the matrix of pairwise comparisons for four different criteria by $\tilde{a}_{12} = (3, 4, 5)$ as a TFN; $\tilde{a}_{14} = (7, 8, 9)_{\delta=3, \gamma=3}$ as a TSFN; $\tilde{a}_{31} = (4, 5, 6)$ as a TFN; $\tilde{a}_{32} = (7, 8, 9)_{\delta=3, \gamma=1/3}$ as a TSFN; $\tilde{a}_{34} = (3, 4, 5)$ as a TFN; $\tilde{a}_{42} = (1, 2, 3)_{\delta=1/3, \gamma=3}$ as a TSFN. Table 10 illustrates the priority vector obtained by the proposed method and linear FPP [27]. The last row of Table 10 (SSE) illustrates the amount of Eq. (15) for each method. The amount of SSE for the proposed method is equal to 83.1927 which is less than linear FPP [27]; therefore, the proposed method is superior to the linear FPP [27]. In other words, the proposed method better satisfies the initial judgments compared with linear FPP [27].

Case 2 of Example 8. In order to compare the performance of the proposed method to the linear FPP [27] for dealing with problems in which some of the judgments represented by TSFN, consider a decision maker compares four elements and provides the following judgments; $\tilde{a}_{12} = (1, 2, 3)_{\delta=3, \gamma=1/3}$ as a TSFN; $\tilde{a}_{14} = (7, 8, 9)_{\delta=3, \gamma=3}$ as a TSFN; $\tilde{a}_{31} = (4, 5, 6)_{\delta=3, \gamma=3}$ as a TSFN; $\tilde{a}_{32} = (7, 8, 9)_{\delta=3, \gamma=1/3}$ as a TSFN; $\tilde{a}_{34} = (3, 4, 5)$ as a TFN; $\tilde{a}_{42} = (4, 5, 6)_{\delta=1/3, \gamma=3}$ as a TSFN. Table 11 depicts the priority vector obtained by the proposed method and linear FPP [27]. The amount of SSE for the proposed

Table 11
Priority vectors for Case 2 of Example 8.

	Linear FPP [27]	Proposed method
w_1	0.204663242	0.248183282
w_2	0.101294851	0.072001764
w_3	0.620196189	0.602335674
w_4	0.073845738	0.077479281
SSE	149.4630	127.5801

SSE: sum of squared error.

Table 12
Priority vectors for Case 3 of Example 8.

	Linear FPP [27]	Proposed method
w_1	0.06078316	0.065176983
w_2	0.088252587	0.05141829
w_3	0.626288922	0.556515028
w_4	0.224675336	0.326889699
SSE	167.0810	112.8160

SSE: sum of squared error.

Table 13
Priority vectors for Case 4 of Example 8.

	Linear FPP [27]	Proposed method
w_1	0.170501409	0.263890146
w_2	0.501497982	0.433944917
w_3	0.07619582	0.06688643
w_4	0.251804791	0.235278507
SSE	105.9806	92.3597

SSE: sum of squared error.

method which is equal to 127.5801 is less than linear FPP [27]; therefore, the proposed method better satisfies the initial judgments compared with linear FPP [27].

Case 3 of Example 8. As an another example to illustrate the superiority of the proposed method to linear FPP [27], consider a decision maker provides the matrix of pairwise comparisons for four different criteria by $\tilde{a}_{12} = (3, 4, 5)_{\delta=1/3, \delta=3}$ as a TSFN; $\tilde{a}_{31} = (5, 6, 7)$ as a TFN; $\tilde{a}_{32} = (7, 8, 9)_{\delta=1/3, \gamma=3}$ as a TSFN; $\tilde{a}_{34} = (3, 4, 5)_{\delta=3, \gamma=3}$ as a TSFN; $\tilde{a}_{41} = (7, 8, 9)$ as a TFN; $\tilde{a}_{42} = (4, 5, 6)_{\delta=1/3, \gamma=3}$ as a TSFN. As it demonstrated in Table 12, the amount of SSE for the proposed method is equal to 112.816 which is better than the linear FPP [27] (Table 13).

Case 4 of Example 8. Suppose that a decision maker compares four elements and provides the following judgments $\tilde{a}_{13} = (7, 8, 9)$ as a TFN; $\tilde{a}_{14} = (3, 4, 5)_{\delta=1/3, \delta=3}$ as a TSFN; $\tilde{a}_{21} = (5, 6, 7)_{\delta=3, \delta=3}$ as a TSFN; $\tilde{a}_{23} = (1, 2, 3)_{\delta=1/3, \delta=3}$ as a TSFN; $\tilde{a}_{24} = (3, 4, 5)$ as a TFN; $\tilde{a}_{43} = (7, 8, 9)_{\delta=1/3, \gamma=3}$ as a TSFN. The amount of SSE for the proposed method is equal to 92.3597 which is less than the linear FPP [27]; therefore, the proposed method is superior to the linear FPP [27].

8. Conclusion

This study presents a new method for deriving priorities from fuzzy pairwise comparison judgments based on a new proposed method of simulating fuzzy numbers and a proposed framework based on genetic algorithm. In many of the existing methods which derive fuzzy weights from fuzzy pairwise comparison matrices, due to the large number of fuzzy multiplication, addition and division operations, the resulting fuzzy scores have wide supports and overlap over a large range [18]. As shown by [20,28], the normalization procedure used in some of these methods may even result in irrational final fuzzy scores, where the normalized upper value is less than the normalized mean value, which is less than the normalized

lower value. As it explained in [18], the fuzzy prioritization methods require an additional fuzzy ranking procedure in order to compare the final fuzzy scores. The different ranking procedures, however, often give different ranking results [29]. Therefore, unlike several known methods, the proposed fuzzy prioritization method of this paper derives crisp weights using a proposed modified fuzzy logarithmic least squares model for both consistent and inconsistent pairwise judgments.

The proposed method of deriving priorities considers judgments represented by both triangular and trapezoidal fuzzy numbers against several methods of literature which consider only judgments represented as triangular fuzzy number. Furthermore, this paper is the first study which introduces and considers judgments represented by triangular shaped fuzzy numbers/trapezoidal shaped fuzzy numbers. In fact, since the proposed method of this paper is based on simulating the uncertain judgments, it is invariant to the specific form or the shape of the fuzzy sets. This property provides opportunity for solving prioritization problems with mixed types of comparison judgments (as it is illustrated in Section 7.3), such as fuzzy sets of different forms and intervals.

To illustrate the superiority of the proposed method to the existing methods, it is compared with five well known methods of literature and it is illustrated that the priority vector of the proposed method better satisfies the initial judgments compared with five known methods of literature.

This paper like the five known methods of literature that are compared with the proposed method considers the fuzzy judgments represented as type-1 fuzzy sets. Since type-2 fuzzy sets involve more uncertainties than type-1 fuzzy sets and they provide us with additional degrees of freedom to represent the uncertainties and the fuzziness of the real word, this study can be further extended in future to solve problems with judgments represented as type-2 fuzzy sets.

Acknowledgements

The Author would like to thank the Islamic Azad University, Qazvin branch, Qazvin, Iran for financially supporting this research under Contract No. 91-127.

References

- [1] T.L. Saaty, *Multicriteria Decision Making: The Analytic Hierarchy Process*, RWS Publications, Pittsburgh, 1988.
- [2] O.S. Vaidya, S. Kumar, Analytic hierarchy process: an overview of applications, *Eur. J. Oper. Res.* 169 (1) (2006) 1–29.
- [3] E. Triantaphyllou, C.T. Lin, Development and evaluation of five fuzzy multiattribute decision-making methods, *Int. J. Approx. Reason.* 14 (4) (1996) 281–310.
- [4] E. Herrera-Viedma, F. Herrera, F. Chiclana, M. Luque, Some issues on consistency of fuzzy preference relations, *Eur. J. Oper. Res.* 154 (1) (2004) 98–109.
- [5] M. Fedrizzi, S. Giove, Incomplete pairwise comparison and consistency optimization, *Eur. J. Oper. Res.* 183 (1) (2007) 303–313.
- [6] F. Chiclana, E. Herrera-Viedma, S. Alonso, F. Herrera, A note on the estimation of missing pairwise preference values: a uninorm consistency based method, *Int. J. Uncertain. Fuzziness Knowl. Based Syst.* 16 (2) (2008) 19–32.
- [7] F. Chiclana, E. Herrera-Viedma, S. Alonso, A note on two methods for estimating missing pairwise preference values, *IEEE Trans. Syst. Man Cybern.* 39 (6) (2009) 1628–1633.
- [8] E. Herrera-Viedma, F. Chiclana, F. Herrera, S. Alonso, Group decision-making model with incomplete fuzzy preference relations based on additive consistency, *IEEE Trans. Syst. Man Cybern. B: Cybern.* 37 (1) (2007) 176–189.
- [9] F. Chiclana, E. Herrera-Viedma, S. Alonso, F. Herrera, Cardinal consistency of reciprocal preference relations: a characterization of multiplicative transitivity, *IEEE Trans. Fuzzy Syst.* 17 (1) (2009) 14–23.
- [10] Y.M. Wang, T.M.S. Elhag, Z. Hua, A modified fuzzy logarithmic least squares method for fuzzy analytic hierarchy process, *Fuzzy Sets Syst.* 157 (23) (2006) 3055–3071.
- [11] A.Z. Grzybowski, Note on a new optimization based approach for estimating priority weights and related consistency index, *Expert Syst. Appl.* 39 (2012) 11699–11708.

- [12] T.L. Saaty, Scaling method for priorities in hierarchical structures, *J. Math. Psychol.* 15 (3) (1977) 234–281.
- [13] N.V. Hovananov, J.W. Kolari, M.V. Sokolov, Deriving weights from general pairwise comparison matrices, *Math. Soc. Sci.* 55 (2008) 205–220.
- [14] Y. Dong, Y. Xu, H. Li, M. Dai, A comparative study of the numerical scales and the prioritization methods in AHP, *Eur. J. Oper. Res.* 186 (2008) 229–242.
- [15] A.Z. Grzybowski, Goal programming approach for deriving priority vectors – some new ideas, *Sci. Res. Inst. Math. Comput. Sci.* 1 (9) (2010) 17–27.
- [16] S. Bozóki, J. Fülöp, L. Rónyai, On optimal completion of incomplete pairwise comparison matrices, *Math. Comput. Modell.* 52 (1–2) (2010) 318–333.
- [17] L. Huo, J. Lan, Z. Wang, New parametric prioritization methods for an analytical hierarchy process based on a pairwise comparison matrix, *Math. Comput. Modell.* 54 (11–12) (2011) 2736–2749.
- [18] M.B. Javanbarg, C. Scawthorn, J. Kiyono, B. Shahbodaghkhan, Fuzzy AHP-based multicriteria decision making systems using particle swarm optimization, *Expert Syst. Appl.* 39 (2012) 960–966.
- [19] P.J.M. Van Laarhoven, W. Pedrycz, A fuzzy extension of Saaty's priority theory, *Fuzzy Sets Syst.* 11 (1–3) (1983) 229–241.
- [20] C.G.E. Boender, J.G. De Graan, F.A. Lootsma, Multi-criteria decision analysis with fuzzy pairwise comparisons, *Fuzzy Sets Syst.* 29 (2) (1989) 133–143.
- [21] M. Kwiesielewicz, The logarithmic least squares and the generalized pseudoinverse in estimating ratios, *Eur. J. Oper. Res.* 93 (3) (1996) 611–619.
- [22] M. Kwiesielewicz, A note on the fuzzy extension of Saaty's priority theory, *Fuzzy Sets Syst.* 95 (2) (1998) 161–172.
- [23] J.J. Buckley, Fuzzy hierarchical analysis, *Fuzzy Sets Syst.* 17 (3) (1985) 233–247.
- [24] M. Wagenknecht, K. Hartmann, On fuzzy rank ordering in poly-optimization, *Fuzzy Sets Syst.* 11 (1983) 253–264.
- [25] D.Y. Chang, Applications of the extent analysis method on fuzzy AHP, *Eur. J. Oper. Res.* 95 (3) (1996) 649–655.
- [26] K.J. Zhu, Y. Jing, D.Y. Chang, A discussion on extent analysis method and application of fuzzy AHP, *Eur. J. Oper. Res.* 116 (2) (1999) 450–456.
- [27] L. Mikhailov, Deriving priorities from fuzzy pairwise comparison judgments, *Fuzzy Sets Syst.* 134 (3) (2003) 365–385.
- [28] O. Gogus, T. Boucher, A consistency test for rational weights in multi criterion decision analysis with fuzzy pairwise comparisons, *Fuzzy Sets Syst.* 86 (1997) 129–138.
- [29] G. Bortolan, R. Degani, A review of some methods for ranking fuzzy subsets, *Fuzzy Sets Syst.* 15 (1) (1985) 1–19.
- [30] Z. Xu, J. Chen, Some models for deriving the priority weights from interval fuzzy preference relations, *Eur. J. Oper. Res.* 184 (2008) 266–280.
- [31] J. Wang, J. Lan, P. Ren, Y. Luo, Some programming models to derive priority weights from additive interval fuzzy preference relation, *Knowl. Based Syst.* 27 (2012) 69–77.
- [32] S. Genç, F.E. Boran, D. Akay, Z. Xu, Interval multiplicative transitivity for consistency, missing values and priority weights of interval fuzzy preference relations, *Inform. Sci.* 180 (2010) 4877–4891.
- [33] J. Lan, M. Hu, X. Ye, S. Sun, Deriving interval weights from an interval multiplicative consistent fuzzy preference relation, *Knowl. Based Syst.* 26 (2012) 128–134.
- [34] Z. Xu, Consistency of interval fuzzy preference relations in group decision making, *Appl. Soft Comput.* 11 (2011) 3898–3909.
- [35] T.C. Wang, Y.H. Chen, Applying fuzzy linguistic preference relations to the improvement of consistency of fuzzy AHP, *Inform. Sci.* 178 (19) (2008) 3755–3765.
- [36] I. Millet, The effectiveness of alternative preference elicitation methods in the analytic hierarchy process, *J. Multi-Criteria Decis. Anal.* 6 (1) (1997) 41–51.
- [37] D. Dubois, H. Prade, The three semantics of fuzzy sets, *Fuzzy Sets Syst.* 90 (2) (1997) 141–150.
- [38] J. Bezdek, B. Spillman, R. Spillman, A fuzzy relation space for group decision theory, *Fuzzy Sets Syst.* 1 (4) (1978) 255–268.
- [39] H. Nurmi, Approaches to collective decision making with fuzzy preference relations, *Fuzzy Sets Syst.* 6 (3) (1981) 249–259.
- [40] T. Tanino, Fuzzy preference ordering in group decision making, *Fuzzy Sets Syst.* 12 (2) (1984) 117–131.
- [41] F. Chiclana, F. Herrera, E. Herrera-Viedma, Integrating multiplicative preference relations in a multipurpose decision-making model based on fuzzy preference relations, *Fuzzy Sets Syst.* 122 (2001) 277–291.
- [42] E. Herrera-Viedma, F. Herrera, F. Chiclana, A consensus model for multiperson decision making with different preference structures, *IEEE Trans. Syst. Man Cybern. A: Syst. Hum.* 32 (3) (2002) 394–402.
- [43] E. Herrera-Viedma, F. Javier Cabrerizo, J. Kacprzyk, W. Pedrycz, A review of soft consensus models in a fuzzy environment, *Inform. Fusion* 17 (2014) 4–13.
- [44] J. Kacprzyk, Group decision making with a fuzzy linguistic majority, *Fuzzy Sets Syst.* 18 (2) (1986) 105–118.
- [45] F. Chiclana, J.M. Tapia Garcia, M.J. Del Moral, E. Herrera-Viedma, A statistical comparative study of different similarity measures of consensus in group decision making, *Inform. Sci.* 221 (2013) 110–123.
- [46] T.L. Saaty, *The Analytic Hierarchy Process: Planning, Priority Setting, Resource Allocation*, McGraw-Hill, New York, 1980.
- [47] F. Herrera, E. Herrera-Viedma, J.L. Verdegay, A model of consensus in group decision making under linguistic assessments, *Fuzzy Sets Syst.* 78 (1) (1996) 73–87.
- [48] F. Herrera, E. Herrera-Viedma, J.L. Verdegay, A rational consensus model in group decision making using linguistic assessment, *Fuzzy Sets Syst.* 88 (1) (1997) 31–49.
- [49] J.M. Tapia Garcia, M.J. Del Moral, M.A. Martinez, E. Herrera-Viedma, A consensus model for group decision making problems with linguistic interval fuzzy preference relations, *Expert Syst. Appl.* 39 (11) (2012) 10022–10030.
- [50] J.J. Buckley, E. Eslami, *An Introduction to Fuzzy Logic and Fuzzy Sets (Advances in Intelligent and Soft Computing)*, Physica-Verlag, New York, 2002.
- [51] D. Dubois, H. Prade, *Fuzzy Sets and Systems*, Academic Press, New York, 1980.
- [52] A. Sanayei, S.F. Mousavi, A. Yazdankhah, Group decision making process for supplier selection with VIKOR under fuzzy environment, *Expert Syst. Appl.* 37 (1) (2010) 24–30.
- [53] J. Barzilai, Deriving weights from pairwise comparison matrices, *J. Oper. Res. Soc.* 48 (12) (1997) 1226–1232.
- [54] T.L. Saaty, A scaling method for priorities in hierarchical structure, *J. Math. Psychol.* 15 (1997) 234–281.
- [55] G. Crawford, C. Williams, A note on the analysis of subjective judgment matrices, *J. Math. Psychol.* 29 (1985) 387–405.
- [56] C. Zhou, Z. Lin, C. Liu, Customer-driven product configuration optimization for assemble-to-order manufacturing enterprises, *Int. J. Adv. Manuf. Technol.* 38 (1–2) (2008) 185–194.
- [57] R. Zhang, C. Wu, Bottleneck identification procedures for the job shop scheduling problem with application to genetic algorithms, *Int. J. Adv. Manuf. Technol.* 42 (11–12) (2009) 1153–1164.
- [58] S. Kumanan, G. Jose, K. Raja, Multi-project scheduling using an heuristic and genetic algorithm, *Int. J. Adv. Manuf. Technol.* 31 (3–4) (2006) 360–366.
- [59] H. Ke, W. Ma, J. Ma, Solving project scheduling problem with the philosophy of fuzzy random programming, *Fuzzy Optim. Decis. Mak.* 11 (3) (2012) 269–284.
- [60] A. Duenas, D. Petrovic, Multi-objective genetic algorithm for single machine scheduling problem under fuzziness, *Fuzzy Optim. Decis. Mak.* 7 (1) (2008) 87–104.
- [61] Y. Yun, Hybrid genetic algorithm with adaptive local search scheme, *Comput. Ind. Eng.* 51 (1) (2006) 128–141.
- [62] L.Z. Zadeh, The concept of a linguistic variable and its applications in approximating reasoning, *Inform. Sci.* 8 (1975) 199–251.
- [63] D.E. Goldberg, K. Deb, A comparison of selection schemes used in genetic algorithms, in: *Foundations of Genetic Algorithms 1, FOGA-1*, vol. 1, 1991, pp. 69–93.
- [64] M. Amiri, A. Mohtashami, Buffer allocation in unreliable production lines based on design of experiments, simulation, and genetic algorithm, *Int. J. Adv. Manuf. Technol.* 62 (1–4) (2012) 371–383.
- [65] H.G. Kwak, J. Kim, An integrated genetic algorithm complemented with direct search for optimum design of RC frames, *Comput. Aided Des.* 41 (7) (2009) 490–500.
- [66] A. Wu, H. Yu, S. Jin, K. Lin, G. Schiavone, An incremental genetic algorithm approach to multiprocessor scheduling, *IEEE Trans. Parallel Distrib. Syst.* 15 (9) (2004) 824–834.