Improving Linguistic Pairwise Comparison Consistency via Linguistic Discrete Regions

Hengshan Zhang, Qinghua Zheng, Ting Liu, Zijiang Yang, Minnan Luo, Yu Qu

***Abstract*—Linguistic pairwise comparison matrices are widely used in decision-making procedures. However, the matrices often give conflicting results when there are multiple criteria under consideration. Despite intensive research, achieving consistency of such matrices remains a daunting task. In this paper, a novel approach based on linguistic discrete region is proposed to address the challenge. Unlike existing methods that require a single value for each comparison, our approach allows a comparison to be expressed by a discrete region with multiple linguistic terms. Such front-end gives users more freedom to express their opinions. In the back-end, we propose an iterative searching algorithm that is able to achieve approximate optimal consistency for the comparison matrices with discrete region values. The final results are single-value matrices that not only guarantee approximate optimal consistency but also comply with evaluators' intentions, as our approach does not modify any linguistic values like many existing methods. We have conducted extensive evaluations and our empirical study confirms that the linguistic discrete region based approach significantly improves the consistency of linguistic pairwise comparison matrices.**

***Index Terms*—consistency, linguistic discrete region, pairwise comparison matrix, linguistic term.**

I. Introduction

THE linguistic pairwise comparison, which quantifies comparison of two attributes with a linguistic term [1], is widely used in Multiple Criteria Decision Making ($MCDM$) procedures [2]. The linguistic pairwise comparison matrix becomes inconsistent when the transitivity and reciprocity rules are violated [3]. Formally, a positive reciprocal comparison matrix $A = (a_{ij})_{n\times n}$ is consistent, if $a_{ik}a_{kj} = a_{ij}$, $1 \le i, j, k \le n$ [4]. Since in most cases it is too difficult for evaluators to give a comparison matrix without any inconsistency, Saaty *et al* proposed consistency ratio (CR) [4] to measure the level of inconsistency. CR is defined as $CR = \frac{\lambda_{\max}-n}{(n-1)RI}$, where $\lambda_{\max}$ is the maximal eigenvalue, RI is the average random index based on the matrix size, and n is the order of matrix. It is believed that a matrix with inconsistency is acceptable as long as $CR < 0.1$. Besides CR, other indexes have been proposed to measure the inconsistency[5–18]. For example, Geometric Consistency Index (GCI) [9, 10], $GCI = \frac{2\sum_{i<j}|\log a_{ij} - \log \frac{P_i}{P_j}|}{(n-1)(n-2)}$, where $P_i (1 \le i \le n)$ are the calculated priorities. In our work we adopt CR as it is the most widely used index.

Many approaches have been proposed to fix the inconsistent comparison matrices if the values of their CR are greater than 0.1. These approaches fall into the following categories. Approaches in the first category require manual adjustments until the value of CR becomes less than the threshold. Without automated aid, these approaches often leave the evaluators clueless on how to improve the consistency. In the second category, algorithms are developed to guide the evaluations [19–22]. For example, Ishizaka and Lusti [19] suggested an expert module to improve the consistency of pairwise comparison matrices by detecting rule transgressions, giving hints and suggesting alternatives for discrete values. The problem with such approaches is that the newly suggested values may contradict an evaluator's intention. If this happens, the approach cannot determine which previous decisions cause the conflict and thus should be revised. Approaches in the third category attempt to modify the values in a matrix automatically so that its inconsistency level becomes less than the predefined threshold [23–26]. An obvious issue with these approaches is that the automatically adjusted matrix may no longer respect an evaluator's intention. The last category researchers have extended the *MCDM* framework to accommodate decision values expressed as intervals [27]. There are a number of studies on interval pairwise comparison matrices[3, 27–35]. The problems of inconsistency of such matrices are discussed in [36–39] and many methods have been proposed. Most of which modify the original comparison matrix and thus may contradict evaluators' intentions. In addition, many methods exploit non linear programming models, which are very difficult to solve.

In this paper, we propose an approach that does not require a single linguistic term for a comparison. Instead, a comparison can be quantified by a linguistic discrete region that consists of multiple linguistic terms. Such approach results in a comparison matrix whose elements are linguistic discrete regions. Based on the 2-tuple linguistic model [1, 40, 41], we translate such comparison matrices into set-matrices whose elements are finite sets of real numbers, which are then processed by our iterative searching algorithm. The final results are automatically generated matrices with approximate optimal consistency and all the elements in the final matrices are consistent with the original evaluations.

This work was supported in part by National Science Foundation of China under Grant (91118005, 91218301, 61221063, 61428206), Ministry of Education Innovation Research Team (IRT13035), Key Projects in the National Science and Technology Pillar Program of China (2013BAK09B01), the National Science Foundation (NSF) under grant CCF-1500365 and the Fundamental Research Funds for the Central Universities.

Hengshan Zhang, Qinghua Zheng, Ting Liu, Minnan Luo, and Yu Qu are with the Ministry of Education Key Lab for Intelligent Networks and Network Security, Department of Computer Science and Technology, Xian Jiaotong University, Xian 710049, China (e-mail: hs.zhang@sei.xjtu.edu.cn, qhzheng@mail.xjtu.edu.cn, tingliu@mail.xjtu.edu.cn, minnluo@mail.xjtu.edu.cn, yqu@sei.xjtu.edu.cn).

Zijiang Yang is with Department of Computer Science, Western Michigan University, Kalamazoo, U.S.A. (e-mail: zijiang.yang@wmich.edu)

A first version of the proposed approach has been previously presented in [42, 43]. The main improvements of this study are the discussions of relations between the comparison matrices with linguistic discrete regions and interval comparison matrices, the theoretical proofs and experimental verifications of efficiencies of algorithms in the proposed approach.

The main contributions of the proposed approach are that: 1) It enables the evaluators express their fuzzy intentions with discrete regions rather than specific linguistic term. 2) Coupled with efficient algorithms, such approach leads to matrices that are not only faithful representation of the evaluators' opinions, but also highly consistent. 3) The efficiencies of algorithms in the proposed approach are theoretical proved and experimental verified.

The rest of this paper is organized as follows. After reviewing relevant concepts in Section II, we present our approach in Section III. The empirical study is conducted in section IV. Finally, Section V concludes this paper.

II. Preliminaries

In this section, we review relevant concepts that include the 2-tuple linguistic model and its extension, the hesitant fuzzy linguistic term set, several different types of preference relations and the interval pairwise comparison matrix.

A. 2-tuple Linguistic Model and Its Extension

Let $S = \{s_k | k = 0, 1, \dots, g\}$ be a linguistic term set with the following characteristics:

- S is ordered: $s_i > s_j$ if and only if $i > j$;
- There is a negation operator: Neg (s_i) =s_j, if $j = g - i$.

The 2-tuple linguistic model [1, 40, 41] represents the linguistic information by pairs in the format of (s_i, α_i), where $s_i \in S$ and $\alpha_i \in [-0.5, 0.5)$. A procedure is defined to make transformations between linguistic terms and numerical values.

Definition 1. *[1] Let $\beta \in [0, g]$ be a real number in the granularity interval of the linguistic term set S. Let $i = round(\beta)$ and $\alpha = \beta - i$ be two values such that $i \in [0, g]$ and $\alpha_i \in [-0.5, 0.5)$. Then, α is called a symbolic translation, with round being the usual rounding operation.*

For example, $S = \{s_k | k = 0, 1 \dots, 8\}$. Let $\beta = 5.6, i = round(\beta) = 6, \alpha = 5.6 - 6 = -0.4$. The 2-tuple is $(s_6, -0.4)$.

Definition 2. *[1] Let S be a linguistic term set and $\beta \in [0, g]$ be a value representing the result of a symbolic aggregation operation. The 2-tuple that expresses the information equivalent to β is obtained by the following function: $\Delta : [0, g] \to S \times [-0.5, 0.5)$, $\Delta(\beta) = (s_i, \alpha_i)$, where $i = round(\beta)$ and $\alpha = \beta - i$.*

Clearly, Δ is a one-to-one function with type $[0, g] \to S \times [-0.5, 0.5)$. Let $\bar{S}$ denote the range of Δ. Δ has an inverse function, $\Delta^{-1}(s_i, x) = i + x$.

For example, let $\beta = 4.5 \in [0, 8]$, $i = round(\beta) = round(4.5) = 5$, $\alpha = 4.5 - 5 = -0.5$, $\Delta(\beta) = (s_5, -0.5)$. $\Delta^{-1}(s_5, 0.3) = 5 + 0.3 = 5.3$.

By defining the concept of the numerical scale, Dong *et al.* [44] proposed an extension of the 2-tuple linguistic model to serve the linguistic term sets that are not uniformly and symmetrically distributed.

Definition 3. *[44] Let $S = \{s_i | i = 0, 1, ..., g\}$ be a linguistic term set, and R be a set of real numbers. The function $NS : S \to R$ is defined as a numerical scale of S, and $NS(s_i)$ is called the numerical index of s_i.*

Definition 4. *[44] For any $(s_i, \alpha) \in \bar{S}$, the numerical scale $\overline{NS}$ on $\bar{S}$ is defined by*

$$\overline{NS}((s_i, \alpha)) = \begin{cases} NS(s_i) + \alpha \times (NS(s_{i+1}) - NS(s_i)) & \alpha \geq 0 \\ NS(s_i) + \alpha \times (NS(s_i) - NS(s_{i-1})) & \alpha < 0 \end{cases}$$

Dong *et al.* [45] proposed an interval version to generalize the existing 2-tuple linguistic models.

Definition 5. *[45] Let $M = \{[A_L, A_R] | A_L, A_R \in R, A_L \leq A_R\}$ be a set of interval numbers. The function $INS : S \to M$ is defined as an interval numerical scale of S, and $INS(s_i)$ is called the interval numerical index of s_i.*

Based on the comparison operator presented in Ishibuchi and Tanaka [46], the interval numerical scale is ordered if $INS_L(s_i) < INS_L(s_{i+1})$ and $INS_R(s_i) < INS_R(s_{i+1})$, for $i = 0, 1, \dots, g-1$. Where, $INS_L(s_i) = A_L^i$, $INS_R(s_i) = A_R^i$.

Definition 6. *[45] For any $(s_i, \alpha) \in \bar{S}$, the interval numerical scale $\overline{INS}$ on $\bar{S}$ is defined by $\overline{INS}((s_i, \alpha)) = [A_L, A_R]$, where*

$$A_L = \begin{cases} INS_L(s_i) + \alpha \times (INS_L(s_{i+1}) - INS_L(s_i)) & \alpha \geq 0 \\ INS_L(s_i) + \alpha \times (INS_L(s_i) - INS_L(s_{i-1})) & \alpha < 0 \end{cases}$$

$$A_R = \begin{cases} INS_R(s_i) + \alpha \times (INS_R(s_{i+1}) - INS_R(s_i)) & \alpha \geq 0 \\ INS_R(s_i) + \alpha \times (INS_R(s_i) - INS_R(s_{i-1})) & \alpha < 0 \end{cases}$$

B. Hesitant Fuzzy Linguistic Term Set

Based on the fuzzy linguistic approach, the hesitant fuzzy linguistic term set (*HFLTS*) will increase the flexibility of the elicitation of linguistic information by means of linguistic expressions.

Definition 7. *[47] Let S be a linguistic term set. A hesitant fuzzy linguistic term set (HFLTS) HS is an ordered finite subset of the consecutive linguistic terms of S.*

- *Empty $HFLTS : HS(\vartheta) = \emptyset$,*
- *Full $HFLTS : HS(\vartheta) = S$.*

A non-empty $HFLTS$ contains at least one linguistic term in S. As discussed in the following definition, a context-free grammar GH provides a way to generate linguistic terms and linguistic expressions.

Definition 8. *[47] Let $GH = (VN, VT, I, P)$ be a context-free grammar, and $S = \{s_0, s_1, \dots, s_g\}$ be a linguistic term set. The elements of GH are defined as follows: $VN = \{< primary\ term >, < composite\ term >, < unary\ relation >, < binary\ relation >, < conjunction >\}, VT = \{lower\ than, greater\ than, between, s_0, s_1, \dots, s_g\}, I \in VN$. P is the production rules that are defined in an extended Backus-Naur form [48].*

Note that $< unary\ relation >$ has some limitations. If the nonterminal symbol is $lower\ than$, the $primary\ term$ cannot be s_0. Similarly, if the nonterminal symbol is $greater\ than$, the $primary\ term$ cannot be s_g.

Definition 9. *[47] Let $E_{GH} : LE \rightarrow H_S$ be a function that transforms linguistic expressions (LE) into $HFLTS(H_S)$. LE is obtained by GH, and S is the linguistic term set that is used by GH.*

The linguistic expressions will be transformed into $HFLTS$ in different ways according to the following rules:

- $E_{GH}(s_i) = \{s_i|s_i \in S\}$;
- $E_{GH}(less\ than\ s_i) = \{s_j|s_j \in S\ and\ s_j \leq s_i\}$;
- $E_{GH}(greater\ than\ s_i) = \{s_j|s_j \in S\ and\ s_j \geq s_i\}$;
- $E_{GH}(between\ s_i\ and\ s_j) = \{s_k|s_k \in S\ and\ s_i \leq s_k \leq s_j\}$.

C. Linguistic Preference Relations and Numerical Preference Relations

As discussed in previous researches, there are two types of preference relations: linguistic preference relations [49–52] and numerical preference relations [16, 53–61]. Let $X = \{x_1, x_2, ..., x_n\}(n \geq 2)$ be a finite set of attributes. When an evaluator makes pairwise comparisons using the linguistic term set *S*, she can construct a linguistic preference relation $L = (l_{ij})_{n\times n}$, in which l_{ij} estimates the preference degree of attribute x_i over x_j.

Definition 10. *[49, 50] The matrix $L = (l_{ij})_{n\times n}$ is called a simple linguistic preference relation if $l_{ij} \in S$, and is called a 2-tuple linguistic preference relation if $l_{ij} \in \bar{S}$. L is considered reciprocal if $l_{ij} = Neg(l_{ji})$ $(1 \leq i, j \leq n)$. The negation operator is defined as the following: $Neg(l_{ij}) = Neg((s_i, \alpha)) = \Delta(g - \Delta^{-1}(s_i, \alpha))$.*

Two widely used numerical preference relations are fuzzy and multiplicative preference relations.

Definition 11. *[16, 54, 55, 62–64] The matrix $F = (f_{ij})_{n\times n}$, where $f_{ij} \in [0, 1]$ and $f_{ij} + f_{ji} = 1$ for $i, j = 1, 2, \ldots, n$, is called a fuzzy preference relation.*

Definition 12. *[29] The matrix $A = (a_{ij})_{n\times n}$, such that $a_{ij} > 0$ and $a_{ij} \times a_{ji} = 1$ for $i, j = 1, 2, \ldots, n$, is called a multiplicative preference relation.*

Ramík and Vlach [8] prove that the multiplicative and fuzzy preference relations can be transformed to each other under different preference representation structures, and the consistency indexes for them are equivalent.

D. Interval Pairwise Comparison Matrix

An interval pairwise comparison matrix can be represented as $\bar{A} = ([a^l_{ij}, a^h_{ij}])_{n\times n}$, where $0 < a^l_{ij} \leqslant a^h_{ij}$, $a^l_{ji} = \frac{1}{a^h_{ij}}$, and $a^h_{ji} = \frac{1}{a^l_{ij}}$.

Definition 13. *[33]. Let $\bar{A} = ([a^l_{ij}, a^h_{ij}])_{n\times n}$ be a comparison matrix. If the convex feasible region $S_w = \{w = (w_1, \ldots, w_n)\, | a^l_{ij} \leq w_i/w_j \leq a^h_{ij}, \sum_{i=1}^n w_i = 1, w_i > 0\}$ is non-empty, $\bar{A}$ is said to be a consistent interval comparison matrix. Otherwise, $\bar{A}$ is said to be inconsistent.*

TABLE I
LINGUISTIC SCALE

Linguistic term	Linguistic scale
s_0	Absolutely less important (AbL)
s_1	Strongly less important (StL)
s_2	Essentially less important (EsL)
s_3	Weakly less important (WkL)
s_4	Equally important (Eq)
s_5	Weakly more important (Wk)
s_6	Essentially more important (Es)
s_7	Strongly more important (St)
s_8	Absolutely more important (Ab)

Let $w = (w_1, \ldots, w_n)$ be a weight vector, on which two different types of constraints may be imposed. One is the additive constraint $\sum_{i=1}^n w_i = 1$. The other is the multiplicative constraint $\prod_{i=1}^n w_i = 1$, which is equivalent to $\sum_{i=1}^n \ln w_i = 0$.

III. Linguistic Discrete Region Based Evaluation Approach

In the proposed linguistic discrete region based evaluation approach, the evaluators use the linguistic discrete region (multiple continuous linguistic terms) instead of a single linguistic term to quantify evaluations. The obtained comparison matrix with linguistic discrete region is translated into a set-matrix (subsection B) by using the numerical scale. We construct an iterative searching algorithm (*ISA*) to find a comparison matrix with approximate optimal consistency in a set-matrix. The linguistic pairwise comparison matrix that corresponds to the matrix found by *ISA* represents the evaluator's final evaluations. As a result, our approach can improve the consistency of linguistic pairwise comparison matrix without changing the evaluator's intentions. Another iterative searching algorithm is proposed to obtain the matrix with approximate optimal consistency from the interval comparison matrix. The purpose of this algorithm is to empirically study the relationships between the comparison matrix with discrete values and the interval comparison matrix.

A. Linguistic Discrete Region

Rodríguez *et al.* proposed the concept of hesitant fuzzy linguistic term set (*HFLTS*) [47],which keeps the basis on the fuzzy linguistic approach [65] and extends the idea of *HFS* (hesitant fuzzy set) to linguistic context. This concept is very useful in practice. We propose the concept of linguistic discrete region for the evaluators to conveniently represent their judgements based on *HFLTS*.

Definition 14. *Let $S = \{s_k|k = 0, 1, \ldots, g\}$ be a linguistic term set. The linguistic discrete region $D = \ ]s_i, s_j\ ](0 \leq i < j \leq g)$ represents a finite subset of S. That is, $D = \{s_i, s_{i+1}, \ldots, s_j\}$.*

For example, a set of linguistic terms are defined as shown in TABLE I [66], $[s_5, s_7]$ denotes the linguistic scales between weakly important and strongly important. In the traditional pairwise comparison, an evaluator only needs to give the values for the upper triangular matrix because the values in the lower triangular matrix can be inferred. The inference can be applied to matrices with linguistic discrete regions as well. Such inference is based on the concept of symmetrical region defined below.

Definition 15. *Let $D = [s_i, s_j](0 \leq i < j \leq g)$ be a linguistic discrete region. The symmetrical region of D is defined as $\tilde{D} = [s_{g-j}, s_{g-i}] = \{s_{g-j}, \ldots, s_{g-i-1}, s_{g-i}\}$.*

We use software quality evaluation as a case study to illustrate the construction of pairwise comparison matrix using the linguistic discrete regions. The three types of software qualities are "efficiency" (C_1), "reliability" (C_2) and "functionality" (C_3). Assume that an evaluator feels certain that "reliability" is "essentially less important" than "functionality", and the importance of "efficiency" sits between "reliability" and "functionality". However, she is uncertain about the relative scale of "efficiency" compared with other two features.

$$P = \begin{bmatrix} [s_4] & [s_5, s_7] & [s_2, s_3] \\ [s_1, s_3] & [s_4] & [s_2] \\ [s_5, s_6] & [s_6] & [s_4] \end{bmatrix}$$

The evaluations can be summarized as a pairwise comparison matrix P with linguistic discrete regions. The values at (i, j) are the results of comparing C_i against C_j. For example, value s_2 at (2,3) indicates "reliability" is "essentially less important" than "functionality". The linguistic discrete region $[s_5,\ s_7]$ at (1, 2) states that "efficiency" is "weakly", "essentially" or "strongly" more important than "reliability". Similarly, linguistic discrete region $[s_2,\ s_3]$ at (1, 3) states that "efficiency" is "essentially" or "weakly" less important than "functionality". Finally, the lower triangular matrix is inferred following Definition 15.

A consistent comparison matrix P_c can be computed based on the linguistic discrete region comparison matrix P. Since our algorithm does not add any additional values, P_c will not violate the evaluator's intention. We will represent our algorithms in Sections III-D and III-E.

$$P_c = \begin{bmatrix} [s_4] & [s_5] & [s_3] \\ [s_3] & [s_4] & [s_2] \\ [s_5] & [s_6] & [s_4] \end{bmatrix}$$

B. Set-Matrix

The 2-tuple linguistic model [1, 40, 41] and its extension [44] are the popular tools for computing with linguistic scales in decision making. In this paper, we translate the linguistic term set in Table I into the real numbers by using the extension of the 2-tuple linguistic model. As a result, the linguistic discrete region matrix in Section III-A is translated into a multiplicative comparison matrix. In this subsection, we present a new data structure called set-matrix (*SM*) to represent the multiplicative comparison matrix that is translated from the linguistic discrete region matrix.

Definition 16. *Let $S = \{s_0, s_1, \ldots, s_g\}$ be a linguistic term set, $D = [s_i, s_j](0 \leq i \leq j \leq g)$ be a linguistic discrete region, and $f(D)$ be a scale function [66]. The range of $f(D)$, in the format of $[f(s_i), f(s_j)]$ is a ordered set of real number. We named this ordered set as real number discrete region ($RNDR$).*

A set-matrix is represented as $U = (cn_{ij} : [u_{ij}^1, u_{ij}^{cn_{ij}}])_{(n\times n)}$, where $[u_{ij}^1, u_{ij}^{cn_{ij}}]$ is a $RNDR$, and cn_{ij} is its size $(i, j = 1, 2, \ldots, n)$. In [66], the authors demonstrate that the geometrical scale [67] and the *LLM* (Logarithmic Least-squares Method) [9] are the best numerical scale and the best prioritization method, respectively. In this paper, the geometrical scale ($f(s) = (\sqrt{c})^{\Delta^{-1}(s)-4}$, $c = 2$) and the *LLM* are selected; $\alpha = 0$, $\overline{NS}(s_i, 0) = NS(s_i)$. Let $NS(s_i) = f(s_i) = (\sqrt{c})^{(\Delta^{-1}(s_i)-4)}(c = 2)$, $s_i \in S$ can be translated into a real number.

Example 1. *Let P be the pairwise comparison matrix shown in Section III-A. Using the geometrical scale [67], the linguistic discrete region can be translated into the corresponding real number discrete region (RNDR). For example, if a linguistic discrete region is $D = [s_5, s_7]$ where $f(s) = (\sqrt{c})^{\Delta^{-1}(s)-4}$ and $c = 2$, the corresponding real number set is $\{1.4142, 2.000, 2.8284\}$, which can be represented as $[1.4142, 2.8284]$. The matrix P is translated into the following set-matrix.*

$$\begin{bmatrix} 1 : [1.000] & 3 : [1.414, 2.828] & 2 : [0.500, 0.707] \\ 3 : [0.354, 0.707] & 1 : [1.000] & 1 : [0.500] \\ 2 : [1.414, 2.000] & 1 : [2.000] & 1 : [1.000] \end{bmatrix}$$

C. Relations Between Comparison Matrix With Discrete Values and Interval Comparison Matrix

In this section, we discuss the relations between the comparison matrix with discrete values and the interval comparison matrix.

Let $S = \{s_i | i = 0, 1, \ldots, g\}$ be a linguistic term set, and $P = (p_{ij})_{n\times n}$ be a comparison matrix that consists of linguistic discrete regions. Dong *et al.* [37] proposed a model to obtain the interval numerical index ($INS(s_i) = [a_i^l, a_i^h]$) based on the initial numerical index a_i of s_i. The obtained interval numerical index includes the initial numerical index, $a_i \in [a_i^l, a_i^h](i = 0, 1, \ldots, g)$. Let the set of initial numerical indexes for S be $IN(S) = \{a_i | i = 0, 1, \ldots, g\}$, and the set of obtained interval numerical indexes be $\overline{IN(S)} = \{[a_i^l, a_i^h] | i = 0, 1, \ldots, g\}$. The map $F : IN(S) \rightarrow \overline{IN(S)}$ is bijective.

The matrix P can be transformed to a set-matrix $A = (cn_{ij} : [a_{ij}^1, a_{ij}^{cn_{ij}}])_{n\times n}$ or an interval matrix $\bar{A} = ([a_{ij}^l, a_{ij}^h])_{n\times n}$ by using the initial numerical indexes or the interval numerical indexes [37]. If we find a comparison matrix with the approximate optimal consistency in set-matrix A, it corresponds to an interval comparison matrix in $\bar{A}$.

If the linguistic discrete region p_{ij} is translated into a *RNDR* $[a_{ij}^1, a_{ij}^{cn_{ij}}]$ (cn_{ij} is the number of elements) by using the numerical indexes, it can be considered as an interval number. As a result, we can obtain an interval comparison matrix $\widetilde{A} = ([a_{ij}^1, a_{ij}^{cn_{ij}}])_{n\times n}$, where $[a_{ij}^1, a_{ij}^{cn_{ij}}]$ is the interval number. In this paper, we propose two iterative searching algorithms to find a matrix with the approximate optimal consistency in the set-matrix A and the corresponding interval comparison matrix $\widetilde{A}$. Through extensive experiments that are shown in Section IV-E, we confirm that the matrix based on the set-matrix A is very close to the matrix based on the corresponding interval matrix $\widetilde{A}$. The approximate optimal matrix obtained from the set-matrix matches the pairwise comparison matrix in *P*, which does not change evaluators' intentions.

D. Iterative Searching Algorithm

Assuming in a discrete region matrix there are n attributes and the average size of elements is m, there can be as many as $m^{\frac{(n-1)n}{2}}$ different combinations. Apparently, it is very expensive to consider all the combinations. We build a model as follow to mitigate this problem:

$$\begin{cases} \min \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{a_{ij}}{a_{ij}^{*}} \\ s.t.\ \mathrm{A} = (\mathrm{a}_{ij})_{n\times n} \in M_U \\ (a_{ij}^{*})_{n\times n} \in M_n, \quad i,j = 1,2,\cdots,n \end{cases} \tag{1}$$

Where n is the order of matrix, $\frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{a_{ij}}{a_{ij}^{*}}$ denotes the principal eigenvalue of A (Lemma 2), M_U represents the set of the pairwise comparison matrices in the set-matrix U, and M_n is the set of consistent pairwise comparison matrices.

Based on the model (4), we propose an Iterative Searching Algorithm (*ISA*) to obtain pairwise comparison matrices with approximate optimal consistency. Let U be a set-matrix, and $A^{(k)} \in U(k = 0,1,\cdots m)$ be the matrix sequence generated by *ISA*. The prioritization method is the Logarithmic Least-squares Method (*LLM*) [9] in this case study. Let $\omega_i(A^{(k)})$ be the ith row geometrical mean of $A^{(k)}$, we have $\omega_i(A^{(k)}) = (\prod_{j=1}^{n} a_{ij}^{k})^{1/n}$, and $a_{ij}^{(k)*} = \omega_i(A^{(k)})/\omega_j(A^{(k)})$. $A^{(k)*} = (a_{ij}^{(k)*})_{n\times n}$ is called as the weight follow matrix of $A^{(k)}$. The consistency index of matrix $A^{(k)}$ is denoted as $CR(A^{(k)})$.

In the following, we describe the steps of *ISA* as depicted in Fig. 1.

Step 1 Initially $k = 0$, $M = A^{(k)} = (a_{ij}^{(k)})_{n\times n}$ is a matrix whose elements in upper triangular matrix are randomly selected from the corresponding *RNDR* in the upper triangular of the set-matrix U.

Step 2 For matrix $A^{(k)}(k = 0,1,2,\cdots)$, we calculate parameters: $A^{(k)*} = (a_{ij}^{(k)*})_{n\times n}$, and $CR(A^{(k)}) = \frac{\lambda_{\max}(A^{(k)})-n}{(n-1)\times RI}$.

Setp 3 An element in the set-matrix U is a *RNDR* that is represented as $[u_{ij}^{1}, u_{ij}^{cn_{ij}}]$, where $1 \le i,j \le n$, and cn_{ij} is the number of elements in *RNDR*. If there is only one element in *RNDR*, for instance $u_{ij}^{q}(q = 1)$, then $a_{ij}^{(k+1)} = u_{ij}^{q}$. Otherwise, for those elements in the upper triangular of the set-matrix U, if $u_{ij}^{t} \in [u_{ij}^{1}, u_{ij}^{cn_{ij}}]$, and $|\ln u_{ij}^{t} - \ln a_{ij}^{(k)*}|$ is the minimum in *RNDR* $[u_{ij}^{1}, u_{ij}^{cn_{ij}}]$, then $a_{ij}^{(k+1)} = u_{ij}^{t}$. In addition, $a_{ji}^{(k+1)} = 1/a_{ij}^{(k+1)}, a_{ii}^{(k+1)} = 1.000$;

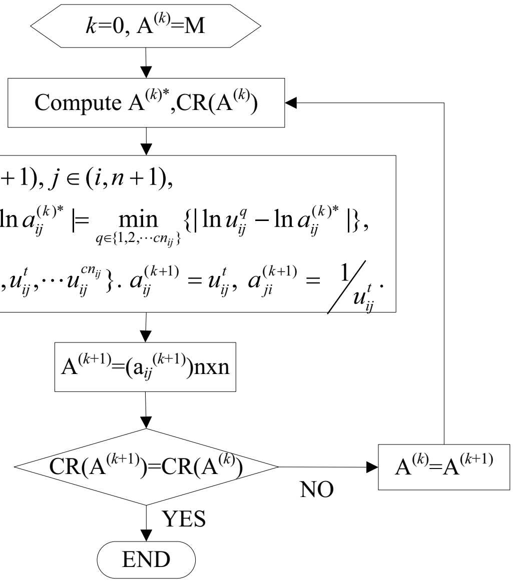

Fig. 1. Flow chart of *ISA*

Step 4 If $CR(A^{(k+1)}) \neq CR(A^{(k)})$, then $A^{(k)}$ is assigned by $A^{(k+1)}$. The algorithm goes back to step 2. Otherwise $CR(A^{(k+1)}) = CR(A^{(k)})$, the algorithm terminates.

If $A^{(k)}(k = 0,1,\cdots m)$ is the matrix sequence computed by *ISA*, we prove that $CR(A^{(k+1)}) \le CR(A^{(k)})$ (Theorem 5 in Section III-F).

ISA returns an approximate optimal comparison matrix from a set-matrix. Although there is computation overhead, our experiments with large number of simulations show that the average number of iterations is less than n (the matrix order).

Example 2. *In this example we illustrate the procedure by applying ISA to the the set-matrix shown in Example 1. An initial matrix* A^0 *is selected from the set-matrix.*

$$A^0 = \begin{bmatrix} 1.000 & 2.000 & 0.500 \\ 0.500 & 1.000 & 0.500 \\ 2.000 & 2.000 & 1.000 \end{bmatrix}$$

$$A^{(0)^*} = \begin{bmatrix} 1.000 & 1.587 & 0.630 \\ 0.630 & 1.000 & 0.400 \\ 1.587 & 2.500 & 1.000 \end{bmatrix}$$

The weight follow matrix ($A^{(0)^*} = (\omega_i^{A^0} / \omega_j^{A^0})_{n\times n}$) *of* A^0 *is computed, and* $CR(A^0)$*=0.05156. The next matrix* $A^1 = (a_{ij}^1)_{n\times n}$ *is obtained from the set-matrix.*

For example, $|\ln 1.587 - \ln 1.414|$ *is the minimum in the set* $\{|\ln 1.587 - \ln 1.414|, \ |\ln 1.587 - \ln 2.000|, \ |\ln 1.587 - \ln 2.828|\}$*, we have* $a_{12}^1 = 1.414$*. Since* $|\ln 0.500 - \ln 0.630| > |\ln 0.707 - \ln 0.630|$*, we have* $a_{13}^1 = 0.707$*. The value at [2,3] in the set-matrix is 0.500, thus* $a_{23}^1 = 0.500$*.*

Repeat this process until the fixed point, i.e. $A^{k+1} = A^k$*, and* A^k *is the final matrix returned by ISA. In this example, the* *optimal matrix A^1 is a consistency matrix, and $CR(A^1)$=0.*

$$A^1 = \begin{bmatrix} 1.000 & 1.414 & 0.707 \\ 0.707 & 1.000 & 0.500 \\ 1.414 & 2.000 & 1.000 \end{bmatrix}$$

E. Iterative Searching Algorithm for Interval Comparison Matrix

Based on the concept of continuous linguistic term set proposed in [68], a linguistic discrete region matrix can be transformed into an interval comparison matrix. In this section, we review the algorithm to compute a matrix with approximate optimal consistency based on a reciprocal interval matrix [43]. We name this algorithm as Iterative Searching Algorithm for Interval Comparison Matrix (*ISAICM*).

Let an interval matrix be $\bar{A} = ([a_{ij}^l, a_{ij}^h])_{n\times n}$, and the matrix sequence generated by *ISAICM* be $A^{(k)} (k = 0, 1, \cdots, m)$. The steps of *ISAICM* are as follows.

Step 1 Choose a comparison matrix $A^{(0)} = (a_{ij}^{(0)})_{n\times n}$ from the given reciprocal interval matrix $\bar{A} = ([a_{ij}^l, a_{ij}^h])_{n\times n}$. The comparison matrix is randomly chosen in this step. For example, $a_{ij}^{(0)} = \frac{a_{ij}^l + a_{ij}^h}{2}, i \le j, (i, j = 1, 2, \cdots, n)$, and $a_{ji}^{(0)} = 1 \Big/ a_{ij}^{(0)}$.

Step 2 For matrix $A^{(k)} (k = 0, 1, 2, \cdots)$, we have: $A^{(k)*} = (a_{ij}^{(k)*})_{n\times n}$, and $CR(A^{(k)}) = \frac{\lambda_{\max}(A^{(k)}) - n}{(n-1)\times RI}$;

Step 3 For $1 \le i \le j \le n$, if $a_{ij}^{(k)*} \in [a_{ij}^l, a_{ij}^h]$, let $a_{ij}^{(k+1)} = a_{ij}^{(k)*}$. If $a_{ij}^{(k)*} < a_{ij}^l$, let $a_{ij}^{(k+1)} = a_{ij}^l$. If $a_{ij}^{(k)*} > a_{ij}^h$, let $a_{ij}^{(k+1)} = a_{ij}^h$. Finally, let $a_{ji}^{(k+1)} = 1 \Big/ a_{ij}^{(k+1)} (i > j)$, and $A^{(k+1)} = (a_{ij}^{(k+1)})_{n\times n}$.

Step 4 Calculate the consistency ratio $CR(A^{(k+1)})$. If $CR(A^{(k+1)}) \ne CR(A^{(k)})$, then $A^{(k)} = A^{(k+1)}$. The procedure goes to step 2. Otherwise, $CR(A^{(k+1)}) = CR(A^{(k)})$, and the algorithm terminates.

If $A^{(k)} (k = 0, 1, 2, \cdots, m)$ is the matrix sequence generated by *ISAICM*, we have $CR(A^{(k+1)}) \le CR(A^{(k)})$. This conclusion has been proved (Theorem 5 in Section III-F). In the following example we illustrate the procedure to compute approximate optimal comparison matrices using *ISAICM*.

Example 3. *The set-matrix given in Example 1 can be considered as an interval matrix:*

$$\bar{A} = \begin{bmatrix} [1.000, 1.000] & [1.414, 2.828] & [0.500, 0.707] \\ [0.354, 0.707] & [1.000, 1.000] & [0.500, 0.500] \\ [1.414, 2.000] & [2.000, 2.000] & [1.000, 1.000] \end{bmatrix}$$

Following is the initial matrix $A^{(0)}$:

$$A^{(0)} = \begin{bmatrix} 1.000 & 2.121 & 0.604 \\ 0.472 & 1.000 & 0.500 \\ 1.656 & 2.000 & 1.000 \end{bmatrix}$$

The weight follow matrix ($A^{(0)^*} = (\omega_i^{A^0} \Big/ \omega_j^{A^0})_{n\times n}$) of A^0 is as the following:

$$A^{(0)^*} = \begin{bmatrix} 1.000 & 1.758 & 0.729 \\ 0.569 & 1.000 & 0.415 \\ 1.372 & 2.413 & 1.000 \end{bmatrix}$$

The next matrix $A^1 = (a_{ij}^1)_{n\times n}$ is computed from the interval matrix. Since $a_{12}^{(0)*} = 1.758$, and $1.758 \in [1.414, 2.828]$, we have $a_{12}^1 = 1.758$. In addition, $a_{13}^{(0)*} = 0.729 \notin [0.500, 0.707]$, and $0.729 > 0.707$. As a result, $a_{13}^1 = 0.707$. $a_{23}^l = a_{23}^h = 0.500$, and $a_{23}^1 = 0.500$.

$$A^1 = \begin{bmatrix} 1.000 & 1.758 & 0.707 \\ 0.569 & 1.000 & 0.500 \\ 1.414 & 2.000 & 1.000 \end{bmatrix}$$

Repeat the procedure until the fixed point $A^{k+1} = A^k$. Following is the approximate optimal matrix A^k returned by ISAICM.

$$A^k = \begin{bmatrix} 1.000 & 1.414 & 0.707 \\ 0.707 & 1.000 & 0.500 \\ 1.414 & 2.000 & 1.000 \end{bmatrix}$$

In [36], Dong $et\ al.$ have proposed a linear programming (*LP*) model,

$$ICI(\bar{A}) = \min_{A \in N_{\bar{A}}} CI\,(A) \qquad (2)$$

Where $\mathrm{CI(A)} = \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} |\log(\mathrm{a}_{ij}) - \log(\omega_i) + \log(\omega_j)|$, $ICI(\bar{A})$ is the consistency index of $\bar{A}$, and $N_{\bar{A}}$ denotes the set of the pairwise comparison matrices in $\bar{A}$. This model can compute a comparison matrix with approximate optimal consistency in the interval comparison matrix $\bar{A}$. This approximate optimal matrix is equivalent to the one computed by *ISAICM* (see Theorem 8).

We use *ISAICM* and *ISA* to do a large number of random experiments in this paper. In the experiments, it can be observed that the approximate optimal consistency matrices computed by *ISAICM* and *ISA* are very similar.

F. Efficiency of ISA and ISAICM

In this section, we analyse the efficiency of *ISA* and *ISAICM*. Let $R_{M(n)}$ denote the set of the positive reciprocal matrix, where n is the order of matrix. Let $\lambda_{\max}^A$ be the maximal eigenvalue of matrix A. The eigenvector of the eigenvalue is denoted as $\omega_A^T = (\omega_1(A), \omega_2(A), \cdots, \omega_n(A))$, which is also called the prioritization vector. Let $a_{ij}^* = \frac{\omega_i(A)}{\omega_j(A)}$, $\varepsilon_{ij} = \frac{a_{ij}}{a_{ij}^*}$, $A^* = (a_{ij}^*)_{n\times n}$ $(1 \le i, j \le n)$.

Firstly, we prove that the consistencies (CR) of the positive receptacle matrix sequences obtained by *ISA* and *ISAICM* decrease. Secondly, we estimate the deviation between the consistencies of approximate optimal matrices and the optimal result.

1) Decreasing Consistencies.

Theorem 5 shows that the consistencies (CR) of the matrix sequences obtained by *ISA* and *ISAICM* decrease. The proof of the theorem 5 is based on theorems 3-4. Lemma 2 provides a method for computing the maximal eigenvalue.

Lemma 1. *Consider function $f(x) = x + {}^1\!/_x$. If $x > 1$, $f(x)$ is strictly monotonous increased. If $0 < x < 1$, $f(x)$is strictly monotonous decreased. $x = 1$ is the minimum point of function $f(x)$ on (0,1).*

Lemma 2. *Given any matrix* $A=(a_{ij})_{n\times n}\in R_{M(n)}$, $\lambda_{\max}^A=\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{n}\varepsilon_{ij}$ *is the maximal eigenvalue of matrix A.*

Proof. If $\lambda_{\max}^A$ is the maximal eigenvalue of matrix A, and $\omega_A^T=(\omega_1(A),\omega_2(A),\cdots,\omega_n(A))$ is its eigenvector, we have $\omega_i(A)>0, i=1,2,\cdots,n$, based on Perron theorem [69]. Furthermore, $\lambda_{\max}^A\omega_i(A)=\sum_{j=1}^{n}a_{ij}\omega_j(A)=\sum_{j=1}^{n}\frac{\omega_i(A)}{\omega_j(A)}\times\varepsilon_{ij}\times\omega_j(A)$
$=\omega_i(A)\sum_{j=1}^{n}\varepsilon_{ij}$, $i=1,2,\cdots,n$.

By removing $\omega_i(A)$ and computing the sum for i, we get $\lambda_{\max}^A=\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{n}\varepsilon_{ij}$ □

Theorem 3. *Let* $A=(a_{ij})_{n\times n}\in R_{M(n)}, B=(b_{ij})_{n\times n}\in R_{M(n)}$ *be two matrices, and* $\mu_{ij}=\frac{b_{ij}}{a_{ij}^*}$.

If $|\ln b_{ij}-\ln a_{ij}^*|\le|\ln a_{ij}-\ln a_{ij}^*|,(i<j \text{ or } i>j)$, *we have:*

(1)$\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{n}\mu_{ij}\le\lambda_{\max}^A$;

(2)$\frac{1}{n}\sum_{i=1}^{n}\sum_{j=i}^{n}|\ln b_{ij}-\ln a_{ij}^*|\le\frac{1}{n}\sum_{i=1}^{n}\sum_{j=i}^{n}|\ln a_{ij}-\ln a_{ij}^*|$.

If $|\ln b_{ij}-\ln a_{ij}^*|\ge|\ln a_{ij}-\ln a_{ij}^*|,(i<j \text{ or } i>j)$, *we have:*

(3)$\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{n}\mu_{ij}\ge\lambda_{\max}^A$;

(4)$\frac{1}{n}\sum_{i=1}^{n}\sum_{j=i}^{n}|\ln b_{ij}-\ln a_{ij}^*|\ge\frac{1}{n}\sum_{i=1}^{n}\sum_{j=i}^{n}|\ln a_{ij}-\ln a_{ij}^*|$.

Proof. Let $U=(\mu_{ij})_{n\times n}$ and $V=(\varepsilon_{ij})_{n\times n}$, where $\mu_{ij}=b_{ij}/a_{ij}^*,\varepsilon_{ij}=a_{ij}/a_{ij}^*, 1\le i,j\le n$. We have $U\in R_{M(n)}$ and $V\in R_{M(n)}$. If $|\ln b_{ij}-\ln a_{ij}^*|\le|\ln a_{ij}-\ln a_{ij}^*|$ $(i<j$ or $i>j)$, there are four cases depending on the relationships between the values of b_{ij},a_{ij},a_{ij}^*.

1) $b_{ij}\ge a_{ij}^*,a_{ij}\ge a_{ij}^*$: Since $|\ln b_{ij}-\ln a_{ij}^*|\le|\ln a_{ij}-\ln a_{ij}^*|$, we have $\frac{b_{ij}}{a_{ij}^*}\le\frac{a_{ij}}{a_{ij}^*}$ and $a_{ij}\ge b_{ij}\ge a_{ij}^*\Rightarrow 1\le\mu_{ij}=\frac{b_{ij}}{a_{ij}^*}\le\varepsilon_{ij}=\frac{a_{ij}}{a_{ij}^*}$. Based on lemma 1, $\mu_{ij}+\frac{1}{\mu_{ij}}\le\varepsilon_{ij}+\frac{1}{\varepsilon_{ij}}$.
2) $b_{ij}\le a_{ij}^*,a_{ij}\le a_{ij}^*$: Based on the known conditions, we have $\frac{a_{ij}^*}{b_{ij}}\le\frac{a_{ij}^*}{a_{ij}}$, and $b_{ij}\ge a_{ij}\Rightarrow a_{ij}\le b_{ij}\le a_{ij}^*\Rightarrow 1\ge\mu_{ij}=\frac{b_{ij}}{a_{ij}^*}\ge\varepsilon_{ij}=\frac{a_{ij}}{a_{ij}^*}$. According to Lemma 1, $\mu_{ij}+\frac{1}{\mu_{ij}}\le\varepsilon_{ij}+\frac{1}{\varepsilon_{ij}}$.
3) $b_{ij}\ge a_{ij}^*,a_{ij}\le a_{ij}^*$: Based on the known conditions, we have $\frac{b_{ij}}{a_{ij}^*}\le\frac{a_{ij}^*}{a_{ij}}$, and $1\le\mu_{ij}=\frac{b_{ij}}{a_{ij}^*}\le\frac{1}{\varepsilon_{ij}}=\frac{a_{ij}^*}{a_{ij}}$. According to Lemma 1, $\mu_{ij}+\frac{1}{\mu_{ij}}\le\varepsilon_{ij}+\frac{1}{\varepsilon_{ij}}$.
4) $b_{ij}\le a_{ij}^*,a_{ij}\ge a_{ij}^*$: Based on the known conditions, we have $\frac{a_{ij}^*}{b_{ij}}\le\frac{a_{ij}}{a_{ij}^*}\Rightarrow\varepsilon_{ij}\ge\frac{1}{\mu_{ij}}\ge 1$. According to Lemma 1, $\mu_{ij}+\frac{1}{\mu_{ij}}\le\varepsilon_{ij}+\frac{1}{\varepsilon_{ij}}$.

For matrices U and V, by computing the sum to i,j, we obtain $\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{n}\mu_{ij}\le\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{n}\varepsilon_{ij}$.

If $|\ln b_{ij}-\ln a_{ij}^*|\le|\ln a_{ij}-\ln a_{ij}^*|$ $(i<j)$, then $|\ln b_{ij}-\ln a_{ij}^*|\le|\ln a_{ij}-\ln a_{ij}^*|$ $(i>j)$. We obtain $\frac{1}{n}\sum_{i=1}^{n}\sum_{j=i}^{n}|\ln b_{ij}-\ln a_{ij}^*|\le\frac{1}{n}\sum_{i=1}^{n}\sum_{j=i}^{n}|\ln a_{ij}-\ln a_{ij}^*|$.

If $|\ln b_{ij}-\ln a_{ij}^*|\ge|\ln a_{ij}-\ln a_{ij}^*|(i<j \text{ or } i>j)$, the proofs of equations (3) and (4) are similar to equations (1) and (2). □

Theorem 4. *Given two matrices* $A=(a_{ij})_{n\times n}\in R_{M(n)}$ *and* $B=(b_{ij})_{n\times n}\in R_{M(n)}$, *we have* $\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{n}\frac{b_{ij}}{b_{ij}^*}\le\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{n}\frac{b_{ij}}{a_{ij}^*}$, *if the method that derives the priority vector from the numerical pairwise comparison matrix is Logarithmic Least-squares Method.*

Proof. According to Logarithmic Least-squares Method [9], we have

$$\begin{cases}\min\sum_{i=1}^{n}\sum_{j>i}[\ln b_{ij}-(\ln\omega_i(B)-\ln\omega_j(B))]^2\\ s.t.\omega_i(B)>0,\sum_{i=1}^{n}\omega_i(B)=1\end{cases}$$

Let $S=\sum_{i=1}^{n}\sum_{j>i}[\ln b_{ij}-(\ln\omega_i(B)-\ln\omega_j(B))]^2$. If S is the minimum at point ${\omega_i}^0(B)(1\le i\le n)$, we have that $[\ln(b_{ij})-(ln{\omega_i}^0(B)-ln{\omega_j}^0(B))]^2(i<j,1\le i,j\le n)$ is the minimum. $|\ln(b_{ij})-(ln{\omega_i}^0(B)-ln{\omega_j}^0(B))|(i<j,1\le i,j\le n)$ is also the minimum.

Let $({\omega_1}^0,{\omega_2}^0,\cdots,{\omega_n}^0)$ be the priority vector of matrix B computed based on *LLM* and $b_{ij}^*=\frac{{\omega_i}^0}{{\omega_j}^0}(1\le i,j\le n)$. According to the aforementioned facts, we have that $|\ln(b_{ij})-\ln b_{ij}^*|(j>i,1\le i,j\le n)$ is the minimum. As a result, $|\ln(b_{ij})-\ln b_{ij}^*|\le|\ln(b_{ij})-\ln a_{ij}^*|(1\le i,j\le n)$. Based on theorem 3, we have $\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{n}\frac{b_{ij}}{b_{ij}^*}\le\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{n}\frac{b_{ij}}{a_{ij}^*}$. □

Theorem 5. *Let* $A^{(k)}(k=1,2,\ldots,m)$ *be the matrix sequence generated by ISA or ISAICM, and* $CR(A^{(k)})$ *be the consistency index of* $A^{(k)}$, *we have* $CR(A^{(k+1)})\le CR(A^{(k)})$.

Proof. Let $\lambda_{\max}^{A^{(k)}}(k=1,2,\ldots,m)$ be the maximal eigenvalue of matrix $A^{(k)}$. Based on *ISA* and *ISAICM*, we know: $|\ln a_{ij}^{(k+1)}-\ln a_{ij}^{(k)*}|\le|\ln a_{ij}^{(k)}-\ln a_{ij}^{(k)*}|(i<j)$. According to theorems 3 and 4, we get $\lambda_{\max}^{A^{(k+1)}}\le\lambda_{\max}^{A^{(k)}}\Leftrightarrow CR(A^{(k+1)})\le CR(A^{(k)})$. □

2) Difference Evaluation of Optimal Matrices' Consistencies Computed by *ISA* or *ISAICM* .

Lemma 6. *[69] Given any non negative matrix A, let* $\rho_i(A)=\sum_{j=1}^{n}a_{ij}$, *and* $\rho(A)$ *be the spectral radius of matrix A (the maximal eigenvalue). We have* $\frac{1}{2}\min_i(\rho_i(A)+\rho_i(A^T))\le\rho(A)\le\frac{1}{2}\max_i(\rho_i(A)+\rho_i(A^T))$

Theorem 7. *Let* $E=(e_{ij})_{n\times n}\in R_{M(n)}$ *be the approximate optimal matrix computed by ISA or ISAICM. If* $\forall i,\sum_{j=1}^{n}\frac{e_{ij}}{e_{ij}^*}=\sum_{k=1}^{n}\frac{e_{ki}}{e_{ki}^*}$ *holds, we get:*

1) *the matrix E is the optimal matrix computed by ISA or ISAICM;*

2) *if* $F = (f_{ij})_{n\times n} \in R_{M(n)}$ $(F \neq E$ *) is another approximate optimal matrix computed by ISA or ISAICM, we get* $CR(F) - CR(E) \leq \frac{\delta}{(n-1)\times RI}$, $\delta = (\lambda_{\max}^F - \min_i\{\frac{1}{2}(\sum_{j=1}^{n} \frac{f_{ij}}{f_{ij}^*} + \sum_{k=1}^{n} \frac{f_{ki}}{f_{ki}^*})\})$.

Proof. 1) Let F be another approximate optimum matrix computed by *ISA* or *ISAICM*. For matrix E, we have $|\ln e_{ij} - \ln e_{ij}^*| \leq |\ln f_{ij} - \ln e_{ij}^*|$ $(1 \leq i < j \leq n)$, based on *ISA* or *ISAICM*.

Let $Y = (y_{ij})_{n\times n}(y_{ij} = \frac{f_{ij}}{e_{ij}^*})$, $\omega_E^T = (\omega_1(E), \cdots, \omega_n(E))$, and $D = diag(\omega_1(E), \omega_2(E), \cdots, \ \omega_n(E))$, we obtain $Y = D^{-1}FD \Rightarrow \lambda_{\max}^F = \lambda_{\max}^Y \geq \min_i\{\frac{1}{2}(\sum_{j=1}^{n} y_{ij} + \sum_{k=1}^{n} y_{ki})\}$ (Lemma 6). Based on theorem 3, we have $\min_i\{\frac{1}{2}(\sum_{j=1}^{n} y_{ij} + \sum_{k=1}^{n} y_{ki})\} = \frac{1}{2}(\sum_{j=1}^{n} y_{i_0 j} + \sum_{k=1}^{n} y_{k i_0}) \geq \frac{1}{2}(\sum_{j=1}^{n} \frac{e_{i_0 j}}{e_{i_0 j}^*} + \sum_{k=1}^{n} \frac{e_{k i_0}}{e_{k i_0}^*})$.

According to the known condition, $\forall i, \sum_{j=1}^{n} \frac{e_{ij}}{e_{ij}^*} = \sum_{k=1}^{n} \frac{e_{ki}}{e_{ki}^*}$, we have $\frac{1}{2}(\sum_{j=1}^{n} \frac{e_{i_0 j}}{e_{i_0 j}^*} + \sum_{k=1}^{n} \frac{e_{k i_0}}{e_{k i_0}^*}) = \lambda_{\max}^E$. That is, $\lambda_{\max}^E \leq \lambda_{\max}^F$. Similarly, if the condition, $\forall i, \sum_{j=1}^{n} \frac{f_{ij}}{f_{ij}^*} = \sum_{k=1}^{n} \frac{f_{ki}}{f_{ki}^*}$ is also satisfied by F, we have $\lambda_{\max}^F \leq \lambda_{\max}^E$. As a result, $\lambda_{\max}^E = \lambda_{\max}^F$. This proves that the matrix E is the optimal matrix computed by *ISA* or *ISAICM*.

2) Let $X = (x_{ij})_{n\times n}$ $(x_{ij} = \frac{e_{ij}}{f_{ij}^*})$, we obtain $\lambda_{\max}^E = \lambda_{\max}^X$. Based on lemma 6, we have $\lambda_{\max}^E \geq \min_i\{\frac{1}{2}(\sum_{j=1}^{n} x_{ij} + \sum_{k=1}^{n} x_{ki})\} = \frac{1}{2}(\sum_{j=1}^{n} x_{i_0 j} + \sum_{k=1}^{n} x_{k i_0})$. Based on theorem 3, we have $\frac{1}{2}(\sum_{j=1}^{n} x_{i_0 j} + \sum_{k=1}^{n} x_{k i_0}) \geq \frac{1}{2}(\sum_{j=1}^{n} \frac{f_{i_0 j}}{f_{i_0 j}^*} + \sum_{k=1}^{n} \frac{f_{k i_0}}{f_{k i_0}^*}) \geq \min_i\{\frac{1}{2}(\sum_{j=1}^{n} \frac{f_{ij}}{f_{ij}^*} + \sum_{k=1}^{n} \frac{f_{ki}}{f_{ki}^*})\}$.

If $F \neq E$, $\lambda_{\max}^F - \lambda_{\max}^E \leq \lambda_{\max}^F - \min(\sum_{j=1}^{n} \frac{f_{ij}}{f_{ij}^*} + \sum_{k=1}^{n} \frac{f_{ki}}{f_{ki}^*}) = \delta$. $CR(F) - CR(E) = \frac{\lambda_{\max}^F - \lambda_{\max}^E}{(n-1)\times RI} \leq \frac{\delta}{(n-1)\times RI}$.

□

Theorem 8. *Let $\bar{A}$ be the interval comparison matrix, $C = (c_{ij})_{n\times n}$ be the matrix computed by model (2), and $D = (d_{ij})_{n\times n}$ be the matrix computed by ISAICM. If the approach that derives the priority vector is LLM [9], then C is the matrix computed by ISAICM and D is a solution of model (2).*

Proof. Based on the model (2), we have

$$ICI(\bar{A}) = \min_{A \in N_{\bar{A}}} CI\,(A) = CI(C) \qquad (3)$$

where $CI(C) = \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n} |lnc_{ij} - ln\omega_i(C) + ln\omega_j(C)|$. Equation (3) shows that $|lnc_{ij} - ln\omega_i(C) + ln\omega_j(C)| = |lnc_{ij} - lnc_{ij}^*|$ is the minimum for $1 \leq i, j \leq n$. Based on theorems 3 and 4, we can conclude that $\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{n} \frac{c_{ij}}{c_{ij}^*}$ is the approximate minimal one for all the comparison matrices in $\bar{A}$. As a result, *C* is the matrix computed by *ISAICM*.

Let $\bar{A} = ([a_{ij}^l, a_{ij}^h])_{n\times n}$. If $d_{ij}^* \in [a_{ij}^l, a_{ij}^h]$, according to *ISAICM*, $d_{ij}^* = d_{ij}$. If $d_{ij}^* \notin [a_{ij}^l, a_{ij}^h]$, $d_{ij} = a_{ij}^l$ or $d_{ij} = a_{ij}^h$. These two cases can ensure that $|ln(d_{ij}^*) - ln(d_{ij})|(i, j = 1, 2, \ldots, n)$ is the minimum in $\bar{A}$. As a result, *D* is a solution of model (2). □

IV. Experiments

In this section we represent experimental results that illustrate the effectiveness of the proposed approach.

A. Traditional Pairwise Comparison and Linguistic Discrete Region Pairwise Comparison

We invited several evaluators to evaluate eight features of the relational database systems that are selected from the *ISO/IEC* 25010 software quality model. The features include functionality, reliability, efficiency, operability, security, compatibility, maintainability and portability. All evaluators are asked to give their evaluations twice: a traditional pairwise comparison and a linguistic discrete region pairwise comparison. We randomly choose evaluators *C* to show the whole process.

1) Traditional Pairwise Comparison

Let the traditional pairwise comparison matrix given by evaluators C be C_1.

The comparison matrix C_1 can be translated into a real number matrix through the 2-tuple linguistic model. The consistency index of C_1 is $CR = 0.028708$.

$$C_1 = \begin{bmatrix} s_4 & s_3 & s_2 & s_6 & s_1 & s_1 & s_3 & s_3 \\ & s_4 & s_5 & s_7 & s_5 & s_6 & s_7 & s_7 \\ & & s_4 & s_7 & s_3 & s_5 & s_7 & s_7 \\ & & & s_4 & s_2 & s_2 & s_4 & s_3 \\ & & & & s_4 & s_6 & s_7 & s_7 \\ & & & & & s_4 & s_4 & s_4 \\ & & & & & & s_4 & s_4 \\ & & & & & & & s_4 \end{bmatrix}$$

2) Linguistic Discrete Region Pairwise Comparison

Let the linguistic discrete region pairwise comparison matrix given by evaluators *C* be C_2.

Since C_1 and C_2 are given by the same evaluators, the linguistic discrete regions in C_2 usually contain the corresponding values in C_1. After C_2 is translated into a set-matrix U, the procedure of the approach is as follows.

First iteration:

Step-1-1: Choose a matrix *M* from the set-matrix C_2. The elements of upper triangular of M are the maximums in related *RNDR*s of C_2.

1.000	0.707	0.500	2.000	0.707	0.707	0.707	0.707
1.414	1.000	1.414	2.828	1.414	2.000	2.828	2.828
2.000	0.707	1.000	2.828	1.000	2.828	2.828	4.000
0.500	0.354	0.354	1.000	0.707	0.500	1.000	1.000
1.414	0.707	1.000	1.414	1.000	2.000	2.828	2.828
1.414	0.500	0.354	2.000	0.500	1.000	1.414	1.414
1.414	0.354	0.354	1.000	0.354	0.707	1.000	1.000
1.414	0.354	0.250	1.000	0.354	0.707	1.000	1.000

$$C_2 = \begin{bmatrix} [s_4] & [s_1,s_3] & [s_0,s_2] & [s_5,s_6] & [s_0,s_3] & [s_1,s_3] & [s_1,s_3] & [s_1,s_3] \\ & [s_4] & [s_4,s_5] & [s_5,s_7] & [s_3,s_5] & [s_4,s_5] & [s_5,s_7] & [s_5,s_7] \\ & & [s_4] & [s_6,s_7] & [s_3,s_4] & [s_5,s_7] & [s_5,s_7] & [s_6,s_8] \\ & & & [s_4] & [s_1,s_3] & [s_0,s_2] & [s_3,s_4] & [s_3,s_4] \\ & & & & [s_4] & [s_4,s_6] & [s_5,s_7] & [s_5,s_7] \\ & & & & & [s_4] & [s_4,s_5] & [s_4,s_5] \\ & & & & & & [s_4] & [s_2,s_4] \\ & & & & & & & [s_4] \end{bmatrix}$$

Step-1-2: Calculate the matrix $A^{(k)*}$ and the consistency index CR of matrix $A^{(k)}$

$$\begin{bmatrix} 1.000 & 0.439 & 0.439 & 1.297 & 0.545 & 0.878 & 1.189 & 1.242 \\ 2.278 & 1.000 & 2.000 & 1.954 & 2.242 & 2.000 & 2.708 & 2.828 \\ 2.278 & 1.000 & 1.000 & 2.954 & 1.242 & 2.000 & 2.708 & 2.828 \\ 0.771 & 0.339 & 0.339 & 1.000 & 0.420 & 0.677 & 0.917 & 0.958 \\ 1.824 & 0.805 & 0.805 & 2.378 & 1.000 & 1.611 & 2.181 & 2.278 \\ 1.139 & 0.500 & 0.500 & 1.477 & 0.621 & 1.000 & 1.354 & 1.414 \\ 0.841 & 0.369 & 0.369 & 1.091 & 0.459 & 0.738 & 1.000 & 1.044 \\ 0.805 & 0.354 & 0.354 & 1.044 & 0.439 & 0.707 & 0.958 & 1.000 \end{bmatrix}$$

$CR(A^k) = 0.027050.$

Step-1-3: Based on $A^{(k)*}$, the next matrix $A^{(k+1)}$ is computed according to set-matrix C_2.

$$\begin{bmatrix} 1.000 & 0.500 & 0.500 & 1.414 & 0.500 & 0.707 & 0.707 & 0.707 \\ 2.000 & 1.000 & 1.000 & 2.828 & 1.414 & 2.000 & 2.828 & 2.828 \\ 2.000 & 1.000 & 1.000 & 2.828 & 1.000 & 2.000 & 2.828 & 2.828 \\ 0.707 & 0.354 & 0.354 & 1.000 & 0.354 & 0.500 & 0.707 & 0.707 \\ 2.000 & 0.707 & 1.000 & 2.828 & 1.000 & 1.414 & 2.000 & 2.000 \\ 1.414 & 0.500 & 0.500 & 2.000 & 0.707 & 1.000 & 1.414 & 1.414 \\ 1.414 & 0.354 & 0.354 & 1.414 & 0.500 & 0.707 & 1.000 & 1.000 \\ 1.414 & 0.354 & 0.354 & 1.000 & 0.500 & 0.707 & 1.000 & 1.000 \end{bmatrix}$$

Step-1-4: The value of $CR(A^{(k+1)})$ is 0.008031. Since $CR(A^{(k+1)}) \neq CR(A^{(k)})$, the algorithm starts the second iteration.

Second iteration:

Step-2-1: Calculate the matrix $A^{(k)*}$ and the consistency index CR of matrix $A^{(k)}$

$$\begin{bmatrix} 1.000 & 0.386 & 0.403 & 1.189 & 0.479 & 0.707 & 1.000 & 1.000 \\ 2.594 & 1.000 & 1.044 & 3.084 & 1.242 & 1.834 & 2.594 & 2.594 \\ 2.484 & 0.958 & 1.000 & 2.954 & 1.189 & 1.756 & 2.484 & 2.484 \\ 0.841 & 0.324 & 0.339 & 1.000 & 0.403 & 0.595 & 0.841 & 0.841 \\ 2.089 & 0.805 & 0.841 & 2.484 & 1.000 & 1.477 & 2.089 & 2.089 \\ 1.414 & 0.545 & 0.569 & 1.682 & 0.677 & 1.000 & 1.414 & 1.414 \\ 1.000 & 0.386 & 0.403 & 1.189 & 0.479 & 0.707 & 1.000 & 1.000 \\ 1.000 & 0.386 & 0.403 & 1.189 & 0.479 & 0.707 & 1.000 & 1.000 \end{bmatrix}$$

$CR(A^k) = 0.008031.$

Step-2-2: Based on $A^{(k)*}$, the next matrix $A^{(k)}$ from the set-matrix C_2 is obtained.

$$\begin{bmatrix} 1.000 & 0.354 & 0.354 & 1.414 & 0.500 & 0.707 & 0.707 & 0.707 \\ 2.828 & 1.000 & 1.000 & 2.828 & 1.414 & 2.000 & 2.828 & 2.828 \\ 2.828 & 1.000 & 1.000 & 2.828 & 1.000 & 2.000 & 2.828 & 2.828 \\ 0.707 & 0.354 & 0.354 & 1.000 & 0.354 & 0.500 & 0.707 & 0.707 \\ 2.000 & 0.707 & 1.000 & 2.828 & 1.000 & 1.414 & 2.000 & 2.000 \\ 1.414 & 0.500 & 0.500 & 2.000 & 0.707 & 1.000 & 1.414 & 1.414 \\ 1.414 & 0.354 & 0.354 & 1.414 & 0.500 & 0.707 & 1.000 & 1.000 \\ 1.414 & 0.354 & 0.354 & 1.414 & 0.500 & 0.707 & 1.000 & 1.000 \end{bmatrix}$$

Step-2-3: The value of $CR(A^{(k+1)})$ is 0.004576. Since $CR(A^{(k+1)}) \neq CR(A^{(k)})$, the algorithm starts the third iteration.

Third iteration:

Step-3-1: Calculate the matrix $A^{(k)*}$ and the consistency index CR of matrix $A^{(k)}$

$$\begin{bmatrix} 1.000 & 0.339 & 0.354 & 1.189 & 0.439 & 0.648 & 0.878 & 0.878 \\ 2.954 & 1.000 & 1.044 & 3.513 & 1.297 & 1.915 & 2.594 & 2.594 \\ 2.828 & 0.958 & 1.000 & 3.364 & 1.242 & 1.834 & 2.484 & 2.484 \\ 0.841 & 0.285 & 0.297 & 1.000 & 0.369 & 0.545 & 0.738 & 0.738 \\ 2.278 & 0.771 & 0.805 & 2.709 & 1.000 & 1.477 & 2.000 & 2.000 \\ 1.542 & 0.522 & 0.545 & 1.834 & 0.677 & 1.000 & 1.354 & 1.354 \\ 1.139 & 0.386 & 0.403 & 1.354 & 0.500 & 0.738 & 1.000 & 1.000 \\ 1.139 & 0.386 & 0.403 & 1.354 & 0.500 & 0.738 & 1.000 & 1.000 \end{bmatrix}$$

$CR(A^k) = 0.004576.$

Step-3-2: Based on $A^{(k)*}$, the next matrix $A^{(k+1)}$ from the set-matrix C_2 is obtained.

$$\begin{bmatrix} 1.000 & 0.354 & 0.354 & 1.414 & 0.500 & 0.707 & 0.707 & 0.707 \\ 2.828 & 1.000 & 1.000 & 2.828 & 1.414 & 2.000 & 2.828 & 2.828 \\ 2.828 & 1.000 & 1.000 & 2.828 & 1.000 & 2.000 & 2.828 & 2.828 \\ 0.707 & 0.354 & 0.354 & 1.000 & 0.354 & 0.500 & 0.707 & 0.707 \\ 2.000 & 0.707 & 1.000 & 2.828 & 1.000 & 1.414 & 2.000 & 2.000 \\ 1.414 & 0.500 & 0.500 & 2.000 & 0.707 & 1.000 & 1.414 & 1.414 \\ 1.414 & 0.354 & 0.354 & 1.414 & 0.500 & 0.707 & 1.000 & 1.000 \\ 1.414 & 0.354 & 0.354 & 1.414 & 0.500 & 0.707 & 1.000 & 1.000 \end{bmatrix}$$

Step-3-3: The value of $CR(A^{(k+1)})$ is 0.004576. Since $CR(A^{(k+1)}) = CR(A^{(k)})$, the algorithm terminates. $A^{(k)}$ is an approximate optimal matrix. Its consistency index is less than the corresponding consistency index of matrix C_1.

3) Performance Analysis

The consistency indexes of all the comparison matrices are given in TABLE II. It shows that the consistency indexes (CR) of the linguistic discrete region comparison matrices are lower than those of the traditional comparison matrices. In most cases, the differences are significant.

B. Random Experiment 1

We consider the positive receptacle matrices with orders 5-16, 20, and 50. For each matrix order, 100 random set-matrices are generated. Each simulates the set-matrix produced by the linguistic discrete region pairwise comparison. For each set-matrix, we run *ISA* 500 times. An approximate optimal matrix is obtained and its consistency is calculated in each run of *ISA*. For each set-matrix, there are 500 approximate optimal consistency indexes. The average and standard deviation of these values, the maximal and minimal consistent indexes are computed.

Fig. 2 depicts the experimental results of the matrices with order 8. The results of the others matrices with different orders are similar. The horizontal axis gives the serial numbers of the set-matrices. The vertical axis gives the values of standard deviation for the consistency indexes (CR) and the differences between the maximal and minimal consistency indexes. The standard deviation of consistency indexes (CR) are very small,

TABLE II
CONSISTENCIES OF THE MATRICES FOR ALL EVALUATORS

	1	2	3	4	5	6	7	8
traditional	0.015328	0.024706	0.009562	0.019957	0.028708	0.019200	0.018513	0.011102
discrete region	0.004960	0.003809	0.000000	0.004191	0.003048	0.004195	0.006874	0.003826

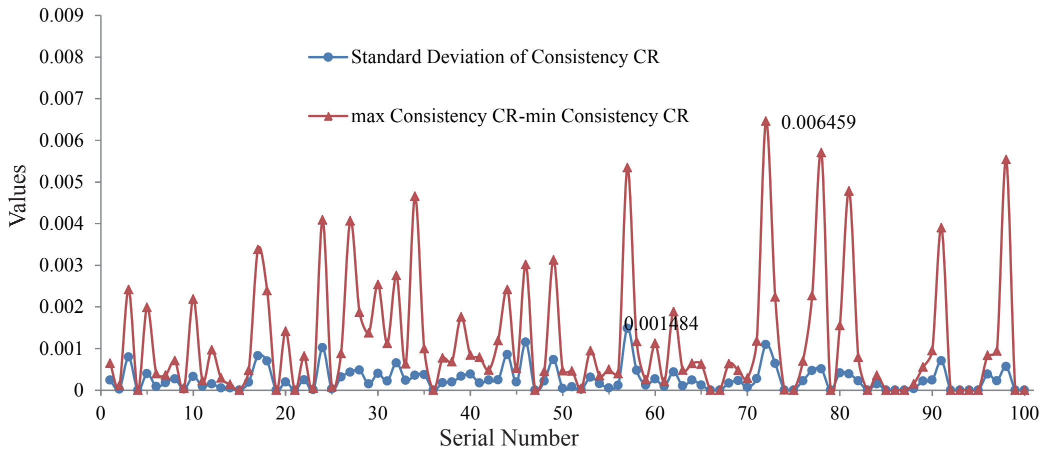

Fig. 2. Standard deviations of CRs and maximal differences of max CRs and min CRs for all set-matrices.

with 0.001484 and 0 being the maximal and minimal values for all the set-matrices. Fig. 2 gives the differences between the maximal and minimal consistency indexes CR. The largest difference is 0.006459 for 100 set-matrices. For all the 5-16 order matrices, the average numbers of iterations of *ISA* are listed in TABLE IV.

C. Random Experiment 2

Again we consider the positive receptacle matrices with orders 5-16, 20, and 50. For every order, a consistent positive reciprocal matrix is generated. This special matrix is called the Optimal Seed Matrix (*OSM*). Based on *OSM*, 100 random set-matrices are randomly generated. There is an *OSM* in each set-matrix. For each set-matrix, we run *ISA* 200 times. In each run, the average, the standard deviation, the maximal and minimal of the consistency indexes for the approximate optimal matrices are calculated.

Fig. 3 depicts the experimental results of matrices with order 7. The horizontal axis shows the serial numbers of the set-matrix, and the vertical axis gives the values of consistency index CR. The maximal value of the standard deviation is 0.001938. It shows that the consistency indexes of approximate optimal matrices obtained by *ISA* are very close to each other. For all the set-matrices, the minimal and maximal consistency indexes of the approximate optimal matrices are 0 and 0.009067, respectively. It shows that the difference between the optimal and approximate optimal consistency index is less than 0.009067 in these experiments.

In the experiments, a final weight vector is also calculated when *ISA* computes an approximate optimal matrix according to a set-matrix. This weight vector is called the Weight of *ISA* (*WOISA*). In addition, the weight vector of *OSM* is named as the Weight of Optimal Seed Matrix (*WOSM*). In each experiment, the cosine similarity of *WOSM* and *WOISA* is calculated. For all the 200 experiments of each set-matrix, the average, the maximum, the minimum, and the standard deviation of the cosine similarities are computed. The experimental results of 7-order matrices are depicted in Fig. 4 and 5. In Fig.4, the maximal standard deviation of the cosine similarities between *WOSM*s and *WOISA*s is 0.00401 among all the set-matrices. It shows that the cosine similarities between *WOSM*s and *WOISA*s for all experiments are very close to each other. In Fig. 5, the average of the cosine similarities between *WOSM*s and *WOISA*s is very close to 1. The minimal average of the cosine similarities between *WOSM*s and *WOISA*s is 0.994729.

In summary, both *WOSM*s and *WOISA*s are very similar in all the experiments. It shows that the weight vector computed based on the approximate optimal matrix is very similar to the weight vector of the optimal matrix in the set-matrix.

D. Random Experiment 3

In this experiment, the range of matrix orders is as before. For each matrix order, 100 linguistic discrete region comparison matrices are randomly generated and then translated into the interval comparison matrices. For each interval comparison matrix, we run *ISAICM* 500 times. In each run of *ISAICM*, an approximate optimal matrix is obtained, its consistency (*CR*) and weight vector are also calculated. We treat the first calculated weight vector as the initial weight vector, and compute the cosine similarities between this initial weight vector and other weight vectors. The maximal differences, the standard deviations of these consistencies are calculated, the minimal and averaging values of cosine similarities are also calculated. Fig. 6 and 7 give the experimental results of order seven matrices. It can be observed that the maximal

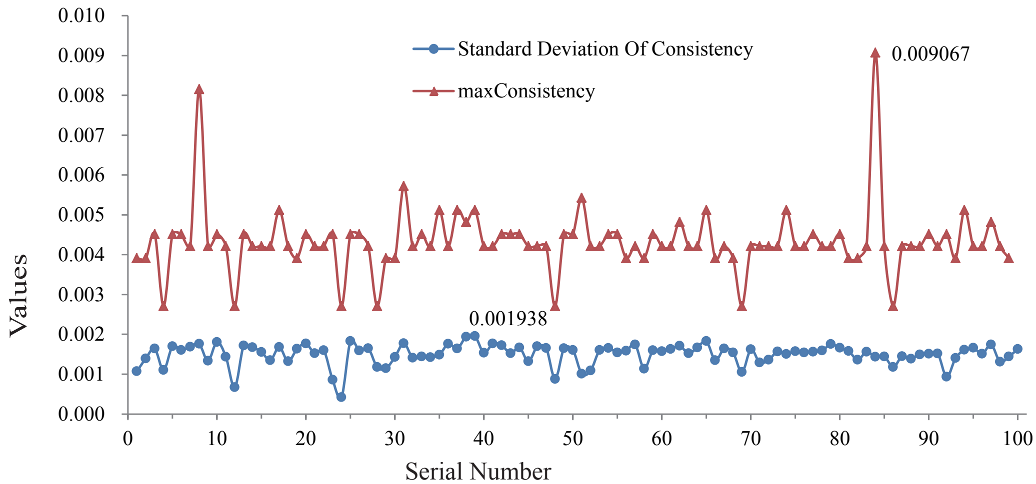

Fig. 3. Standard deviations and maximal CRs for all set-matrices.

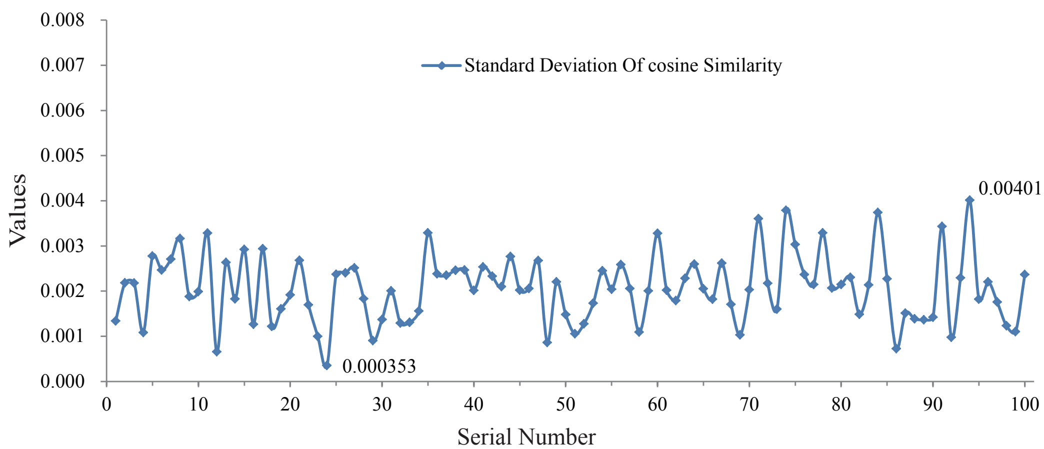

Fig. 4. Standard deviations of cosine similarities of weight vectors.

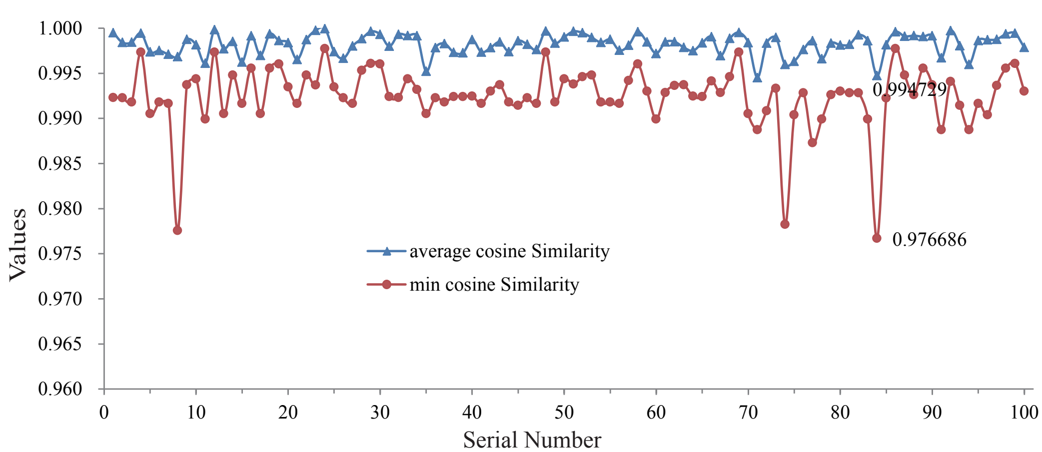

Fig. 5. Average and minimal cosine similarities.

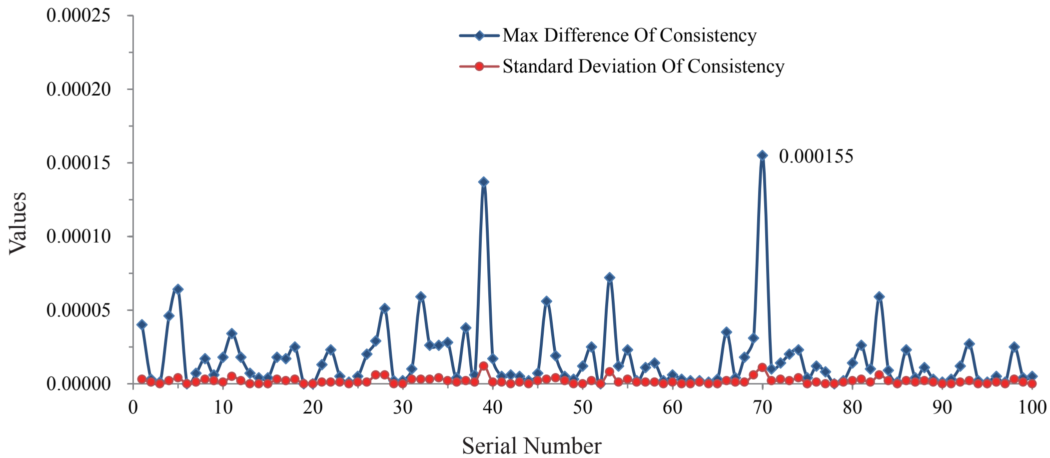

Fig. 6. Maximal differences and standard deviations of consistency indexes.

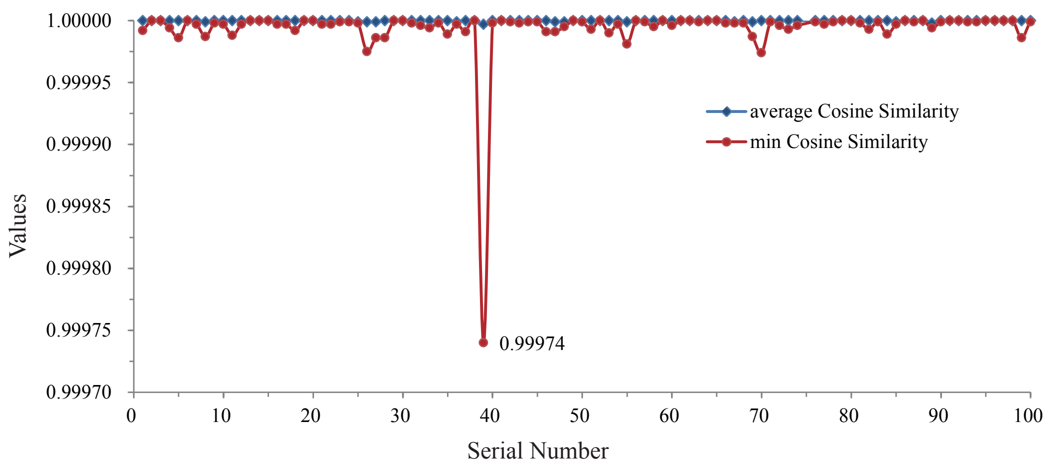

Fig. 7. Average and minimal cosine similarities.

difference and standard deviation are 0.000155 and 0.000012, respectively (Fig.6). It shows that the consistencies obtained by *ISAICM* are very close to each other. The minimal cosine similarity of the weight vectors computed by *ISAICM* is 0.99974. All the average cosine similarities are close to 1.0 (Fig.7). It shows that the weight vectors computed by *ISAICM* are very similar to each other. The results of the matrices with other orders are very similar.

E. Random Experiment 4

In this experiment, 1000 random linguistic comparison matrices are generated for each order, and the orders of matrices range from 5 to 16. Each linguistic comparison matrix is transformed into a set-matrix and an interval reciprocal matrix, and *ISA* and *ISAICM* are used to compute the approximate optimal matrices. The initial comparison matrices are randomly chosen from the set-matrices (*ISA*) and interval comparison matrices (*ISAICM*). The consistency indices *CR* and the weight vectors are computed based on the approximate optimal matrices.

The experimental results are shown in TABLE III and TABLE IV. In Table III, "*CR Differ.*" refers to the differences between *CR*s computed by *ISA* and *ISAICM*, and "Weight Similarity" represents the cosine similarities between two weight vectors derived by *ISA* and *ISAICM*. For each order, we report its average and maximal values. It can be observed that the maximal difference between *CR*s is 0.01313. The average of cosine similarities between two weight vectors derived by *ISA* and *ISAICM* is greater than 0.999, the minimal value is greater than 0.97 for all experiments. The average numbers of iterations of the two algorithms are shown in TABLE IV. The matrix orders range from 5 to 16. It can be observed that the numbers of iterations are relatively small, even less than the order of matrices.

V. Conclusion and Future Work

In this paper, we have proposed a new approach that is able to improve the consistency of linguistic pairwise comparison matrices. Our approach allows the evaluators to express their fuzzy intentions with linguistic discrete regions rather than specific values. We have designed two algorithms (*ISA* and *ISAICM*) to compute the approximate optimal matrices based on the set-matrices and interval comparison matrices. The weight vectors computed using *ISA* and *ISAICM* are very similar. The approximate optimal matrices computed by *ISA*

TABLE III
THE DIFFERENCES OF CR AND THE COSINE SIMILARITIES OF WEIGHT VECTORS FOR ISA AND ISAICM

order	type	CR Differ.	Weight Similarity	order	type	CR Differ.	Weight Similarity	order	type	CR Differ.	Weight Similarity
5	Avg.	0.00134	0.99923	9	Avg.	0.00065	0.99979	13	Avg.	0.00052	0.99990
5	Max.	0.01313	0.97289	9	Max.	0.00328	0.98969	13	Max.	0.00162	0.99907
6	Avg.	0.00101	0.99958	10	Avg.	0.00064	0.99984	14	Avg.	0.00048	0.99992
6	Max.	0.01044	0.98942	10	Max.	0.00449	0.99778	14	Max.	0.00178	0.99891
7	Avg.	0.00084	0.99964	11	Avg.	0.00057	0.99986	15	Avg.	0.00048	0.99993
7	Max.	0.00563	0.98800	11	Max.	0.00208	0.99864	15	Max.	0.00142	0.99953
8	Avg.	0.00072	0.99976	12	Avg.	0.00051	0.99990	16	Avg.	0.00000	0.99993
8	Max.	0.00582	0.99554	12	Max.	0.00254	0.99889	16	Max.	0.00000	0.99912

TABLE IV
THE AVERAGE NUMBERS OF ITERATIONS FOR TWO ALGORITHMS

order	5	6	7	8	9	10	11	12	13	14	15	16
ISA	4.05	4.91	5.47	6.29	6.82	7.82	8.23	9.18	9.79	10.24	10.59	11.00
ISAICM	5.94	5.78	5.65	5.49	5.27	5.23	5.07	4.90	4.78	4.74	4.66	4.59

are not only a faithful representation of evaluators' intentions, but also highly consistent.

Furthermore, by using the matrix theory, we have proved that these algorithms are efficient. The experimental results also confirm that the proposed approach has good performance. In the future, we will consider other consistency indexes. For example, 2-tuple linguistic index [5] and the ordinal consistency index (OCI)[18]. We believe *ISA* and *ISAICM* can be adapted to improve these consistency indexes as well.

Acknowledgment

The authors would like to thank the Editor, the Associate Editor and the anonymous reviewers for their most constructive comments and suggestions to improve the quality of this paper.

References

[1] F. Herrera and L. Martínez, "A 2-tuple fuzzy linguistic representation model for computing with words," *Fuzzy Systems, IEEE Transactions on*, vol. 8, no. 6, pp. 746–752, 2000.

[2] F. Herrera *et al.*, "An overview on the 2-tuple linguistic model for computing with words in decision making: Extensions, applications and challenges," *Information Sciences*, vol. 207, pp. 1–18, 2012.

[3] F. Liu, "Acceptable consistency analysis of interval reciprocal comparison matrices," *Fuzzy Sets and Systems*, vol. 160, no. 18, pp. 2686–2700, 2009.

[4] T. L. Saaty, "A scaling method for priorities in hierarchical structures," *Journal of mathematical psychology*, vol. 15, no. 3, pp. 234–281, 1977.

[5] Y. Dong, W.-C. Hong, and Y. Xu, "Measuring consistency of linguistic preference relations: a 2-tuple linguistic approach," *Soft Computing*, vol. 17, no. 11, pp. 2117–2130, 2013.

[6] S. Orlovsky, "Decision-making with a fuzzy preference relation," *Fuzzy sets and systems*, vol. 1, no. 3, pp. 155–167, 1978.

[7] T. L. Saaty, "How to make a decision: the analytic hierarchy process," *European journal of operational research*, vol. 48, no. 1, pp. 9–26, 1990.

[8] J. Ramík and M. Vlach, "Measuring consistency and inconsistency of pair comparison systems," *Kybernetika*, vol. 49, no. 3, pp. 465–486, 2013.

[9] G. Crawford and C. Williams, "A note on the analysis of subjective judgment matrices," *Journal of mathematical psychology*, vol. 29, no. 4, pp. 387–405, 1985.

[10] A. Ishizaka and A. Labib, "Review of the main developments in the analytic hierarchy process," *Expert Systems with Applications*, vol. 38, no. 11, pp. 14 336–14 345, 2011.

[11] J. Aguaron and J. M. Moreno-Jiménez, "The geometric consistency index: Approximated thresholds," *European Journal of Operational Research*, vol. 147, no. 1, pp. 137–145, 2003.

[12] J. Peláez and M. Lamata, "A new measure of consistency for positive reciprocal matrices," *Computers & Mathematics with Applications*, vol. 46, no. 12, pp. 1839–1845, 2003.

[13] A. A. Salo and R. P. Hämäläinen, "On the measurement of preferences in the analytic hierarchy process," *Journal of Multi-Criteria Decision Analysis*, vol. 6, no. 6, pp. 309–319, 1997.

[14] P. Ji and R. Jiang, "Scale transitivity in the AHP," *Journal of the Operational Research Society*, vol. 54, no. 8, pp. 896–905, 2003.

[15] J. Ramík and P. Korviny, "Inconsistency of pair-wise comparison matrix with fuzzy elements based on geometric mean," *Fuzzy Sets and Systems*, vol. 161, no. 11, pp. 1604–1613, 2010.

[16] F. Chiclana, E. Herrera-Viedma, S. Alonso, and F. Herrera, "Cardinal consistency of reciprocal preference relations: a characterization of multiplicative transitivity," *Fuzzy Systems, IEEE Transactions on*, vol. 17, no. 1, pp. 14–23, 2009.

[17] G. Zhang, Y. Dong, and Y. Xu, "Consistency and consensus measures for linguistic preference relations based

on distribution assessments," *Information Fusion*, vol. 17, pp. 46–55, 2014.

[18] Y. Xu, R. Patnayakuni, and H. Wang, "The ordinal consistency of a fuzzy preference relation," *Information Sciences*, vol. 224, pp. 152–164, 2013.

[19] A. Ishizaka and M. Lusti, "An expert module to improve the consistency of AHP matrices," *International Transactions in Operational Research*, vol. 11, no. 1, pp. 97–105, 2004.

[20] T. L. Saaty, "Decision-making with the AHP: Why is the principal eigenvector necessary," *European journal of operational research*, vol. 145, no. 1, pp. 85–91, 2003.

[21] D. Cao, L. C. Leung, and J. Law, "Modifying inconsistent comparison matrix in analytic hierarchy process: a heuristic approach," *Decision Support Systems*, vol. 44, no. 4, pp. 944–953, 2008.

[22] T.-C. Wang and Y.-H. Chen, "Applying fuzzy linguistic preference relations to the improvement of consistency of fuzzy AHP," *Information Sciences*, vol. 178, no. 19, pp. 3755–3765, 2008.

[23] D. Ergu, G. Kou, Y. Peng, and Y. Shi, "A simple method to improve the consistency ratio of the pair-wise comparison matrix in ANP," *European Journal of Operational Research*, vol. 213, no. 1, pp. 246–259, 2011.

[24] M. Ghazanfari and M. Nojavan, "Educing inconsistency in fuzzy AHP by mathematical programming models," *Asia-Pacific Journal of Operational Research*, vol. 21, no. 03, pp. 379–391, 2004.

[25] Z. Xu and C. Wei, "A consistency improving method in the analytic hierarchy process," *European Journal of Operational Research*, vol. 116, no. 2, pp. 443–449, 1999.

[26] M. Xia, Z. Xu, and J. Chen, "Algorithms for improving consistency or consensus of reciprocal [0, 1]-valued preference relations," *Fuzzy Sets and Systems*, vol. 216, pp. 108–133, 2013.

[27] Z.-J. Wang and K. W. Li, "Goal programming approaches to deriving interval weights based on interval fuzzy preference relations," *Information Sciences*, vol. 193, pp. 180–198, 2012.

[28] L. Mikhailov, "A fuzzy approach to deriving priorities from interval pairwise comparison judgements," *European Journal of Operational Research*, vol. 159, no. 3, pp. 687–704, 2004.

[29] T. L. Saaty and L. G. Vargas, "Uncertainty and rank order in the analytic hierarchy process," *European Journal of Operational Research*, vol. 32, no. 1, pp. 107–117, 1987.

[30] K. Sugihara, H. Ishii, and H. Tanaka, "Interval priorities in AHP by interval regression analysis," *European Journal of Operational Research*, vol. 158, no. 3, pp. 745–754, 2004.

[31] Y.-M. Wang and T. Elhag, "A goal programming method for obtaining interval weights from an interval comparison matrix," *European Journal of Operational Research*, vol. 177, no. 1, pp. 458–471, 2007.

[32] Y.-M. Wang, Z.-P. Fan, and Z. Hua, "A chi-square method for obtaining a priority vector from multiplicative and fuzzy preference relations," *European Journal of Operational Research*, vol. 182, no. 1, pp. 356–366, 2007.

[33] Y.-M. Wang, J.-B. Yang, and D.-L. Xu, "A two-stage logarithmic goal programming method for generating weights from interval comparison matrices," *Fuzzy sets and systems*, vol. 152, no. 3, pp. 475–498, 2005.

[34] Z. Xu, "On compatibility of interval fuzzy preference relations," *Fuzzy Optimization and Decision Making*, vol. 3, no. 3, pp. 217–225, 2004.

[35] Z. Xu and J. Chen, "Some models for deriving the priority weights from interval fuzzy preference relations," *European journal of operational research*, vol. 184, no. 1, pp. 266–280, 2008.

[36] Y. Dong, X. Chen, C.-C. Li, W.-C. Hong, and Y. Xu, "Consistency issues of interval pairwise comparison matrices," *Soft Computing*, vol. 19, pp. 2321–2335, 2015.

[37] Y. Dong and E. Herrera-Viedma, "Consistency-driven automatic methodology to set interval numerical scales of 2-tuple linguistic term sets and its use in the linguistic GDM with preference relation." *IEEE transactions on cybernetics*, vol. 45, no. 5, pp. 780-792, 2015.

[38] Y.-M. Wang, J.-B. Yang, and D.-L. Xu, "Interval weight generation approaches based on consistency test and interval comparison matrices," *Applied Mathematics and Computation*, vol. 167, no. 1, pp. 252–273, 2005.

[39] J. S. Finan and W. J. Hurley, "The analytic hierarchy process: does adjusting a pairwise comparison matrix to improve the consistency ratio help?" *Computers & operations research*, vol. 24, no. 8, pp. 749–755, 1997.

[40] F. Herrera and L. Martínez, "A model based on linguistic 2-tuples for dealing with multigranular hierarchical linguistic contexts in multi-expert decision-making," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 31, no. 2, pp. 227–234, 2001.

[41] F. Herrera and L. Martinez, "The 2-tuple linguistic computational model: advantages of its linguistic description, accuracy and consistency," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 9, no. supp01, pp. 33–48, 2001.

[42] H. Zhang, Q. Zheng, T. Liu, Z. Yang, and J. Liu, "A discrete region-based approach to improve the consistency of pair-wise comparison matrix," in *Fuzzy Systems (FUZZ), 2013 IEEE International Conference on*. IEEE, 2013, pp. 1–7.

[43] H. Zhang, Q. Zheng, T. Liu, and Y. Nan, "A new approach to improve the consistency of linguistic pair-wise comparison matrix and derive interval weight vector," in *Fuzzy Systems (FUZZ-IEEE), 2014 IEEE International Conference on*. IEEE, 2014, pp. 1190–1196.

[44] Y. Dong, Y. Xu, and S. Yu, "Computing the numerical scale of the linguistic term set for the 2-tuple fuzzy linguistic representation model," *Fuzzy Systems, IEEE Transactions on*, vol. 17, no. 6, pp. 1366–1378, 2009.

[45] Y. Dong, G. Zhang, W.-C. Hong, and S. Yu, "Linguistic computational model based on 2-tuples and intervals," *IEEE Transactions on fuzzy systems*, vol. 21, no. 6, pp. 1006–1018, 2013.

[46] H. Ishibuchi and H. Tanaka, "Multiobjective program-

ming in optimization of the interval objective function," *European Journal of Operational Research*, vol. 48, no. 2, pp. 219–225, 1990.
[47] R. M. Rodriguez, L. Martinez, and F. Herrera, "Hesitant fuzzy linguistic term sets for decision making," *Fuzzy Systems, IEEE Transactions on*, vol. 20, no. 1, pp. 109–119, 2012.
[48] G. Bordogna and G. Pasi, "A fuzzy linguistic approach generalizing boolean information retrieval: A model and its evaluation," *JASIS*, vol. 44, no. 2, pp. 70–82, 1993.
[49] S. Alonso, F. Chiclana, F. Herrera, E. Herrera-Viedma, J. Alcalá-Fdez, and C. Porcel, "A consistency-based procedure to estimate missing pairwise preference values," *International Journal of Intelligent Systems*, vol. 23, no. 2, pp. 155–175, 2008.
[50] S. Alonso, F. J. Cabrerizo, F. Chiclana, F. Herrera, and E. Herrera-Viedma, "Group decision making with incomplete fuzzy linguistic preference relations," *International Journal of Intelligent Systems*, vol. 24, no. 2, pp. 201–222, 2009.
[51] F. J. Cabrerizo, E. Herrera-Viedma, and W. Pedrycz, "A method based on PSO and granular computing of linguistic information to solve group decision making problems defined in heterogeneous contexts," *European Journal of Operational Research*, vol. 230, no. 3, pp. 624–633, 2013.
[52] Y. Dong, Y. Xu, and H. Li, "On consistency measures of linguistic preference relations," *European Journal of Operational Research*, vol. 189, no. 2, pp. 430–444, 2008.
[53] M. Brunelli, L. Canal, and M. Fedrizzi, "Inconsistency indices for pairwise comparison matrices: a numerical study," *Annals of Operations Research*, vol. 211, no. 1, pp. 493–509, 2013.
[54] F. Chiclana, F. Herrera, and E. Herrera-Viedma, "Integrating three representation models in fuzzy multipurpose decision making based on fuzzy preference relations," *Fuzzy sets and Systems*, vol. 97, no. 1, pp. 33–48, 1998.
[55] M. Fedrizzi and S. Giove, "Incomplete pairwise comparison and consistency optimization," *European Journal of Operational Research*, vol. 183, no. 1, pp. 303–313, 2007.
[56] E. Herrera-Viedma, F. Herrera, and F. Chiclana, "A consensus model for multiperson decision making with different preference structures," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. 32, no. 3, pp. 394–402, 2002.
[57] J. Kacprzyk, M. Fedrizzi, and H. Nurmi, "Group decision making and consensus under fuzzy preferences and fuzzy majority," *Fuzzy Sets and Systems*, vol. 49, no. 1, pp. 21–31, 1992.
[58] W. Koczkodaj, "A new definition of consistency of pairwise comparisons," *Mathematical and computer modelling*, vol. 18, no. 7, pp. 79–84, 1993.
[59] J. Ma, Z.-P. Fan, Y.-P. Jiang, J.-Y. Mao, and L. Ma, "A method for repairing the inconsistency of fuzzy preference relations," *Fuzzy sets and systems*, vol. 157, no. 1, pp. 20–33, 2006.
[60] B. Srdjevic, "Combining different prioritization methods in the analytic hierarchy process synthesis," *Computers & Operations Research*, vol. 32, no. 7, pp. 1897–1919, 2005.
[61] B. Srdjevic and Z. Srdjevic, "Synthesis of individual best local priority vectors in AHP-group decision making," *Applied soft computing*, vol. 13, no. 4, pp. 2045–2056, 2013.
[62] E. Herrera-Viedma, F. Chiclana, F. Herrera, and S. Alonso, "Group decision-making model with incomplete fuzzy preference relations based on additive consistency," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 37, no. 1, pp. 176–189, 2007.
[63] E. Herrera-Viedma, F. Herrera, F. Chiclana, and M. Luque, "Some issues on consistency of fuzzy preference relations," *European journal of operational research*, vol. 154, no. 1, pp. 98–109, 2004.
[64] T. Tanino, "Fuzzy preference orderings in group decision making," *Fuzzy sets and systems*, vol. 12, no. 2, pp. 117–131, 1984.
[65] L. A. Zadeh, "The concept of a linguistic variable and its application to approximate reasoning," *Information sciences*, vol. 8, no. 3, pp. 199–249, 1975.
[66] Y. Dong, W.-C. Hong, Y. Xu, and S. Yu, "Selecting the individual numerical scale and prioritization method in the analytic hierarchy process: A 2-tuple fuzzy linguistic approach," *Fuzzy Systems, IEEE Transactions on*, vol. 19, no. 1, pp. 13–25, 2011.
[67] F. Lootsma, "Scale sensitivity in the multiplicative AHP and SMART," *Journal of Multi-Criteria Decision Analysis*, vol. 2, no. 2, pp. 87–110, 1993.
[68] Y. Dong, Y. Xu, H. Li, and B. Feng, "The OWA-based consensus operator under linguistic representation models using position indexes," *European Journal of Operational Research*, vol. 203, no. 2, pp. 455–463, 2010.
[69] A. H. Roger and R. J. Charles, *matrix analysis*. Cambridge university press, 1990.

Hengshan Zhang received the M.S. degree from the Department of Computer Science and Technology, Lanzhou University, Lanzhou, China, in 2009. He is currently a Ph.D. candidate student at the Ministry of Education Key Lab for Intelligent Networks and Network Security, Xi'an Jiaotong University. His research interests include computing with words, group decision making, information aggregation and trustworthy software.

Qinghua Zheng received the B.S. degree in computer software in 1990, the M.S. degree in computer organization and architecture in 1993, and the Ph.D. degree in system engineering in 1997 from Xi'an Jiaotong University, China. He was a postdoctoral researcher at Harvard University in 2002. He is currently a professor in Xi'an Jiaotong University, and the dean of the Department of Computer Science. His research areas include computer network security, intelligent e-learning theory and algorithm, multimedia e-learning, and trustworthy software.

Ting Liu received his B.S. degree in information engineering and Ph.D. degree in system engineering from School of Electronic and Information, Xi'an Jiaotong University, Xi'an, China, in 2003 and 2010, respectively. Currently, he is an associate professor of the Systems Engineering Institute, Xi'an Jiaotong University. His research interests include Smart Grid, network security and trustworthy software.

Zijiang Yang is an associate professor in computer science at Western Michigan University. He holds a Ph.D. from the University of Pennsylvania, an M.S. from Rice University and a B.S. from the University of Science and Technology of China. Before joining WMU he was an associate research staff member at NEC Labs America. He was also a visiting professor at the University of Michigan from 2009 to 2013. His research interests are in the area of software engineering with the primary focus on the testing, debugging and verification of software systems. He is a senior member of IEEE.

Minnan Luo was born in Shaanxi Province, China, in 1984. She received the Ph. D. degree from the Department of Computer Science and Technology, Tsinghua University, Beijing, China, in 2014. She is currently an Assistant Professor in the School of Electronic and Information Engineering at Xi'an Jiaotong University. Her research interests include fuzzy system identification and modeling, machine learning and optimization and discrete event systems.

Yu Qu received the B.S. degree from the School of Electronic and Information Engineering, Xi'an Jiaotong University, Xi'an, China in 2006. He is currently a Ph.D. candidate student at the Ministry of Education Key Lab for Intelligent Networks and Network Security, Xi'an Jiaotong University. His research interests include trustworthy software and applying complex network and data mining theories to analyzing software systems.