Research Article

A Neural Network-Based Approach to Multi-Attribute Group Decision-Making with Heterogeneous Preference Information

Quan Zhang [ID], Hongyu Li [ID], Xueting Lu [ID], and Chunliu Wu [ID]

School of Artificial Intelligence, ShenYang University of Technology, ShenYang, China

Correspondence should be addressed to Quan Zhang; isqzhang@sut.edu.cn

For the problem of multi-attribute group decision-making with heterogeneous preference information on attribute values and overall preference orderings on alternatives, this article proposes a neural network-based approach. In the approach, firstly, the heterogeneous preference information on attribute values and overall preference orderings on alternatives are normalized. Secondly, based on the normalization results, two optimization models are set up to determine attribute weights and expert weights, respectively. Thirdly, two neural networks are set up and trained to determine attribute weights and expert weights based on the optimization models. Then, the overall values of the alternatives are obtained as well as their rankings. Simulations on the proposed neural networks are conducted for illustrations.

1. Introduction

In the course of multi-attribute decision-making, a group of experts is always invited to take part in the voting work [1–5]. For some complex multi-attribute decision-making problems, for example, the international joint project selections across different countries, the experts may come from different countries. The invited experts participate in ranking and selecting the projects with the help of the Web by providing their evaluation information. They can evaluate the projects against every attribute and can also give their overall opinions on the projects, for example, the preference orderings on the projects. Both the evaluation information of projects on the attributes and the overall preference orderings on the projects are usually applied in ranking and selecting the projects.

The current researches on the multi-attribute group decision-making (MAGDM) problems with preference information fall into two categories, that is, with preference information on attributes and with preference information on alternatives (e.g., the assessment and selection of international joint projects). The preference information can be used to calculate the attribute weights. The determination of attribute weights is an important factor for alternative

selection in multiple attribute group decision-making, which plays a crucial role in the rankings or selection of the alternatives [6, 7]. This article focuses on the MAGDM problems with two sources of information, that is, the preference information on alternatives from the experts and attribute values in the decision matrix. Both sources of information in the MAGDM problems can be used to calculate the attribute weights and determine the alternative rankings based on the experts' collective information.

For the MAGDM problems, the preference information on alternatives can be used to derive their rankings or selection directly [3] or to determine the attribute weights [8, 9]. The attribute values in the decision matrix can also be used to determine the attribute weights, for example, the entropy method [10, 11] and maximizing deviation method [12, 13]. For the MAGDM problems with the preference information on alternatives and decision matrix, it is desirable to establish optimization models to determine attribute weights [8, 14, 15].

Since the research question focused in this study is to integrate the attribute values in the decision matrix with the experts' preference information on alternatives, the neural network is a good choice for memorizing the experts' preference information. However, it is not common to solve

the MAGDM problems with heterogeneous preference information on alternatives by means of neural networks. In this article, two optimization models are established, respectively, to solve the attribute weights and expert weights for the MAGDM problems with heterogeneous preference information on alternatives and decision matrix. In addition, based on the established optimization models, two neural networks are designed to figure out the attribute weights and expert weights. Furthermore, in order to train the neural networks, the attribute values of the alternatives given by the experts are employed as the inputs and the preference information on alternatives from the experts are adopted as the expected outputs. After training the neural networks, the attribute weights are obtained, which match the attribute values of the alternatives with the preference information on alternatives given by the experts. Then, the obtained attribute weights from the neural networks are used for guiding the alternative rankings, which is the novelty of this study. The intention of the research innovation of this study is to apply neural networks in MAGDM problems with heterogeneous preference information on alternatives and decision matrix.

The organization of this article is as follows. Section 2 gives a literature review and motivation. Section 3 describes the MAGDM problems with heterogeneous preference information on decision matrix and alternatives. A neural network-based approach to the MAGDM problems is proposed in Section 4, where the decision information is normalized, and the two optimization models are established. Two neural networks are designed and trained to determine attribute weights and expert weights. Section 5 gives simulations for illustrating the proposed approach. Section 6 summarizes the study in this article.

2. Literature Review and Motivation

2.1. The Preference Information in MAGDM. In the course of multi-attribute group decision-making, based on the experts' preference information, the weights of attributes can be obtained. In [8, 14], the preference information on alternatives is employed to determine the attribute weights by setting up optimization models, respectively.

Also, in the course of multi-attribute group decision-making, based on the experts' preference information, the weights of experts can be obtained. In [5], a new linear programming model is proposed to find the optimal expert weights based on deviation function. In [16], an entropy-based approach is proposed to determine the weights of experts by defining a projection measure for the hybrid information representations.

For the MAGDM problems with heterogeneous preference information on decision matrix and alternatives, it is desirable to establish optimization models to determine attribute weights [8, 14, 15]. In [8], an integration approach is proposed to deal with experts' fuzzy preference information on alternatives and decision matrix by setting up an optimization model. In [14], three sources of information are integrated into a general model framework: experts' fuzzy preference relation on alternatives, experts'

multiplicative preference relation on attributes, and decision matrix. In [15], a consistency-based approach is proposed to the MADM problems with preference information on alternatives, by defining a geometric consistency index. An algorithm is proposed to adjust the preference information and the decision matrix simultaneously to improve the geometric consistency index in MADM.

In this study, two neural networks are employed to memorize the experts' preference information on alternatives and trained by dealing with the attribute values in decision matrix at the same time.

2.2. Research on MAGDM with Preference Based on Neural Networks. Neural networks have the ability of self-adaptive and self-learning and are used in some decision support processes, for example, foreign currency risk prediction [17], demand forecasting [18], travelers' choice patterns recognition [19], and service providers selection [20].

In [17], a neural network is used to predict foreign exchange rate movement direction and magnitude, considering multiple macroeconomic and microstructure of foreign exchange market variables. In [18], a new approach is proposed to forecast the uncertain customer demand in the multilevel supply chain system by means of the neural network method. In [19], a neural network-based approach is developed to investigate the travelers' decision rule heterogeneity, where the neural network is trained to recognize the travelers' choice patterns among four distinct decision rules so that they are classified. In [20], an adaptive fuzzy-neuro approach is proposed for selecting the group of service providers, in which the maintenance service network in agriculture is designed and evaluation criteria are defined from both qualitative and quantitative aspects.

It is noticed that neural networks begin to be applied in multiple attribute decision-making processes, such as [21, 22]. In [21], an approach is proposed to integrate neural networks and data envelopment analysis to evaluate suppliers under incomplete information of evaluation criteria. In [22], a neural network is used to learn the relation among criteria and alternatives and rank the alternatives.

2.3. Motivation of the Study. However, it is not common to capture and represent the preference information of experts by means of neural networks and then assist in solving the MAGDM problems with heterogeneous preference information in an uncertain decision-making environment. It is desirable to employ neural networks to memorize the preference information from the experts and the decision processes by means of training the attribute weights. Also, the neural network is needed for guiding the decision-making processes by memorizing the experts' preference information.

The motivation of this article is to develop and apply neural networks for MAGDM problems with heterogeneous preference information in an uncertain environment and to aid the decision-making task without the experts' efforts further. The necessity of using neural network-based approach in the MAGDM problems is to integrate the attribute

values in the decision matrix with the experts' preference information on alternatives, by means of training the neural networks and memorizing the two sources of information by weights. The contributions of this article are to develop and apply neural networks for memorizing the experts' preference information in MAGDM problems with heterogeneous preference information in an uncertain environment.

3. Problem Descriptions

This article considers the MAGDM problems where the decision matrix is presented with heterogeneous information (i.e., preference orderings [23], interval numbers, linguistics [24, 25], and uncertain linguistic variables [26, 27]) and the invited experts give their overall preference orderings on the alternatives. The following assumptions and notations are used to describe the MAGDM problems.

3.1. Composition of the Problem. This article considers the uncertain group decision problem where the experts provide their decision matrix with heterogeneous preference information (i.e., preference orderings, interval numbers, linguistic terms, and uncertain linguistic variables) and give their overall preference orderings on the alternatives. The following assumptions and notations are used to present the MAGDM problems.

Let $E = \{e^1, e^2, \ldots, e^P\}$ denote the set of experts, who provide preference information on alternatives. On one hand, the experts give their preference information on alternatives against the attributes, respectively. On the other hand, they also express their overall preference orderings on the alternatives.

Let $S = \{S_1, S_2, \ldots, S_m\}$ denote a discrete set of m (>1) possible alternatives.

Let $C = \{C_1, C_2, \ldots, C_n\}$ denote a set of n (>1) attributes.

3.2. Experts' Preference Information. In this article, each expert provides his/her preference information on the alternatives according to each attribute by means of preference orderings, interval numbers, linguistic terms, and uncertain linguistic variables.

Let $\widetilde{A}^k = [\widetilde{a}_{ij}^k]_{m \times n}$ denote the decision matrix given by expert e^k, $k = 1,2,\ldots,P$, where $\widetilde{a}_{ij}^k$ is the preference information on alternative S_i with respect to attribute C_j, $i = 1,2,\ldots,m$, $j = 1,2,\ldots,n$.

Column j of decision matrix $\widetilde{A}^k$ collects preference information on the alternatives according to attribute C_j, and is expressed by means of preference orderings, vectors of interval numbers, vectors of linguistic terms, and vectors of uncertain linguistic variables:

(1) An ordinal ranking $O_j^k = (o_j^k(1), \ldots, o_j^k(m))$ is used by an expert e^k to express his/her preference on the alternatives according to attribute C_j, $j = 1,2,\ldots,n$, where $o_j^k(\cdot)$ is a permutation function over the index set $\{1, \ldots, m\}$ and $o_j^k(i)$ represents the position of alternative S_i against attribute C_j. Usually, the alternatives are ranked from the best to the worst.

(2) A vector of interval numbers $iter_j^k = ([a_{1j}^{kL}, a_{1j}^{kU}], [a_{2j}^{kL}, a_{2j}^{kU}], \ldots, [a_{mj}^{kL}, a_{mj}^{kU}])$ is used by an expert e^k to express his/her preference on the alternatives according to attribute C_j, $j = 1,2,\ldots,n$, where $[a_{ij}^{kL}, a_{ij}^{kU}]$ is an interval number of the preference information on alternative S_i, $i = 1, 2,\ldots, m$.

(3) A vector of linguistic terms $ling_j^k = (ling_{1j}^k, ling_{2j}^k, \ldots, ling_{mj}^k)$ is used by an expert e^k to express his/her preference on the alternatives according to attribute C_j, $j = 1,2,\ldots,n$ [26, 27], where $ling_{ij}^k$ is a linguistic term of the preference information on alternative S_i, $i = 1,2,\ldots,m$. It is assumed that $ling_{ij}^k$ ($i = 1,\ldots,m$) is from the linguistic term set *TERMSET^j* for attribute C_j, and for different attributes, the granularities of *TERMSET^j* may be different, $j = 1, 2,\ldots, n$.

(4) A vector of uncertain linguistic variables $ling_j^k - v = ([ling_{1j}^{kL}, ling_{1j}^{kU}], [ling_{2j}^{kL}, ling_{2j}^{kU}], \ldots, [ling_{mj}^{kL}, ling_{mj}^{kU}])$ is used by an expert e^k to express his/her preference on the alternatives according to attribute C_j, $j = 1,2,\ldots,n$, $k = 1,2,\ldots,P$, where $[ling_{ij}^{kL}][ling_{ij}^{kU}]$ is an uncertain linguistic variable of the preference information on alternative S_i against attribute C_j, $i = 1,2,\ldots,m$, $j = 1,2,\ldots,n$, $k = 1,2,\ldots,P$. It is assumed that $ling_{ij}^{kL}$ and $ling_{ij}^{kU}$ ($i = 1,\ldots,m$, $j = 1,2,\ldots,n$, $k = 1,2,\ldots,P$) are from the linguistic term set *TERMSET^j* for attribute C_j, and for different attributes, the granularities of *TERMSET^j* may be different, $j = 1, 2,\ldots,n$.

In addition, the experts also express their overall preference on the alternatives in the form of preference orderings. Suppose $R^k = (r_1^k, r_2^k, \ldots, r_m^k)$ is given by expert e^k, $k = 1,2,\ldots,P$ and r_i^k denotes the overall preference ordering of alternative S_i, $i = 1, 2,\ldots, m$.

`The problem focused is to rank the alternatives and select the best one based on the preference information on the attribute values (i.e., $\widetilde{A}^k = [\widetilde{a}_{ij}^k]_{m \times n}$, $k = 1,2,\ldots,P$) and the overall preference orderings on the alternatives (i.e., R^k, $k = 1,2,\ldots,P$).

In the following section, a new approach to the problem is proposed by setting up optimization models and neural networks. The heterogeneous preference information on the alternatives against the attributes, that is, $\widetilde{A}^k = [\widetilde{a}_{ij}^k]_{m \times n}$, $k = 1,2,\ldots,P$, are normalized. In the meantime, the experts' overall preference orderings on the alternatives, that is, R^k, $k = 1,2,\ldots,P$, are normalized too. Based on the normalization results, two optimization models are set up and neural networks are designed to obtain the attribute weights and expert weights, based on which the weighted sum method is employed to calculate the overall values of the alternatives.

3.3. Focus of the Problem. The problem focused is to rank the alternatives and select the best one based on the preference information on the heterogeneous attribute values in the decision matrix and the overall preference orderings on the alternatives. In the following section, a neural network-based approach is proposed, where the heterogeneous

preference information on the attribute values is normalized. Two optimization models are set up and accordingly, two neural networks are designed and trained to obtain the attribute weights and expert weights. Finally, the rankings of the alternatives are obtained.

The neural networks proposed in this study are intended to solve the MAGDM problems with heterogeneous preference information on the alternatives, by means of their learning abilities, which can memorize the subjective preference information given by experts and further guide the solution to the MAGDM problems.

4. The Proposed Approach

In the proposed approach, four steps are employed.

Step 1. Normalize the heterogeneous attribute values in the decision matrix and the overall preference orderings on the alternatives.

Step 2. Set up optimization models to determine attribute weights and expert weights.

Step 3. Set up neural networks to determine attribute weights and expert weights based on the optimization models set up in step 2.

Step 4. Calculate the overall values of the alternatives.

4.1. Normalize Decision Information.
In the decision matrix $A^k = [a_{ij}^k]_{m \times n}$ given by expert e^k ($k = 1, \ldots, P$), a_{ij}^k can be expressed in the forms of preference orderings, interval numbers, linguistic terms, and linguistic variables for different j, $1 \leq j \leq n$, and therefore normalization on a_{ij}^k are necessary to make them comparable (i.e., the utilities values between 0 and 1), denote the normalization result of $A^k = [a_{ij}^k]_{m \times n}$ as $B^k = [b_{ij}^k]_{m \times n}$, $k = 1, \ldots, P$.

4.1.1. Normalize Preference Orderings.
For the attributes which are evaluated by means of preference orderings, that is, $O_j = (o^j(1), \ldots, o^j(m),)$ for C_j, the following method can be applied to normalize it [1]:

$$pr_{il}^j = \frac{1}{2}\left(1 + \frac{o^j(l) - o^j(i)}{m - 1}\right), i, l = 1, \ldots, m, j = 1, \ldots, n. \tag{1}$$

Then, the attribute value of alternative S_i can be obtained:

$$z_i^j = \frac{1}{m(m-1)}\left(\sum_{l=1}^m pr_{il}^j + \frac{m}{2} - 1\right), \tag{2}$$

$$i = 1, \ldots, m,$$

$$j = 1, \ldots, n.$$

Furthermore, the attribute value of alternative S_i can be normalized by the following way:

$$b_{ij} = \frac{\left|z_i^j - \min_{1 \leq l \leq m}\{z_l^j\}\right|}{\left|z_i^j - \min_{1 \leq l \leq m}\{z_l^j\}\right| + \left|z_i^j - \max_{1 \leq l \leq m}\{z_l^j\}\right|}, \tag{3}$$

$$i = 1, \ldots, m,$$

$$j = 1, \ldots, n.$$

4.1.2. Normalize Intervals.
Firstly, normalize the intervals $a_{ij} = [a_{ij}^L, a_{ij}^U]$ into dimensionless ones by means of the following methods.

For beneficial attributes:

$$\% \quad \begin{cases} q_{ij}^L = \dfrac{a_{ij}^L}{\sum_{l=1}^m a_{lj}^U}, \\[3mm] q_{ij}^U = \dfrac{a_{ij}^U}{\sum_{l=1}^m a_{lj}^L}, \end{cases} \tag{4}$$

$$i = 1, \ldots, m,$$

$$j = 1, \ldots, n.$$

For the costly attributes:

$$\begin{cases} q_{ij}^L = \dfrac{1/a_{ij}^U}{\sum_{l=1}^m 1/a_{ij}^L}, \\[3mm] q_{ij}^U = \dfrac{1/a_{ij}^L}{\sum_{l=1}^m 1/a_{ij}^U}, \end{cases} \tag{5}$$

$$i = 1, \ldots, m,$$

$$j = 1, \ldots, n.$$

Then transform $q_{ij} = [q_{ij}^L, q_{ij}^U]$ into the crisp values, and obtain $B = [b_{ij}]_{m \times n}$.

Definition 1. Given intervals q_{ij} and q_{kj}, i, $k = 1, \ldots, m$, $j = 1, \ldots, n$, the distance between them is defined as:

$$\text{dev}\left(q_{ij}, q_{kj}\right) = \frac{\sqrt{2}}{2}\sqrt{\left(q_{ij}^L - q_{kj}^L\right)^2 + \left(q_{ij}^U - q_{kj}^U\right)^2}, i, k$$

$$= 1, \ldots, m, j = 1, \ldots, n. \tag{6}$$

Definition 2. Given the attribute values of intervals for attribute C_j, the positive ideal attribute value for it is defined as:

$$\text{pos}_j^+ = \left[\text{pos}_j^{+L}, \text{pos}_j^{+U}\right], j = 1, \ldots, n, \tag{7}$$

where

$$\text{pos}_j^{+L} = \max_{1 \leq i \leq m}\{q_{ij}^L\}, j = 1, \ldots, n,$$

$$\text{pos}_j^{+U} = \max_{1 \leq i \leq m}\{q_{ij}^U\}, j = 1, \ldots, n. \tag{8}$$

Definition 3. Given the attribute values of intervals for attribute C_j, the negative ideal attribute value for it is defined as:

$$\text{neg}_j^- = \left[\text{neg}_j^{-L}, \text{neg}_j^{-U}\right], j = 1, \ldots, n, \tag{9}$$

where

$$\text{neg}_j^{-L} = \min_{1 \le i \le m} \left\{q_{ij}^L\right\}, j = 1, \ldots, n,$$
$$\text{neg}_j^{-U} = \min_{1 \le i \le m} \left\{q_{ij}^U\right\}, j = 1, \ldots, n. \tag{10}$$

Then the distance between attribute value $q_{ij} = [q_{ij}^L, q_{ij}^U]$ and pos_j^+ and neg_j^- are obtained as follows:

$$\text{dev}\left(q_{ij}, \text{pos}_j^+\right) = \frac{\sqrt{2}}{2} \sqrt{\left(q_{ij}^L - \text{pos}_j^{+L}\right)^2 + \left(q_{ij}^U - \text{pos}_j^{+U}\right)^2}, i$$

$$= 1, \ldots, m, j = 1, \ldots, n,$$

$$\text{dev}\left(q_{ij}, \text{neg}_j^-\right) = \frac{\sqrt{2}}{2} \sqrt{\left(q_{ij}^L - \text{neg}_j^{-L}\right)^2 + \left(q_{ij}^U - \text{neg}_j^{-U}\right)^2}, i$$

$$= 1, \ldots, m, j = 1, \ldots, n. \tag{11}$$

Definition 4. Given the attribute values of intervals for attribute C_j, the relative distance between q_{ij} and neg_j^- is defined as:

$$b_{ij} = \frac{\text{dev}\left(q_{ij}, \text{neg}_j^-\right)}{\text{dev}\left(q_{ij}, \text{pos}_j^+\right) + \text{dev}\left(q_{ij}, \text{neg}_j^-\right)}, i$$

$$= 1, \ldots, m, j = 1, \ldots, n. \tag{12}$$

4.1.3. Normalize Linguistic Information. Linguistic information is very helpful for experts to express their opinions in uncertain decision-making situations. For different attributes, linguistic terms employed for assessment tasks are often of different granularities.

Suppose $TERMSET = \{l_0, l_1, \ldots, lg\}$ is a linguistic term set with $(g + 1)$ elements. By definitions in [28], a linguistic term set is orderly and there are inverse operators.

Definition 5. If a linguistic term l_i is expressed in the form of triangular fuzzy numbers $(\alpha_i, \beta_i, \gamma_i)$, then its membership function is [28]:

$$\mu_i(x) = \begin{cases} \dfrac{x - \alpha_i}{\beta_i - \alpha_i}, & \alpha_i < x < \beta_i, \\ \\ \dfrac{x - \beta_i}{\gamma_i - \beta_i}, & \beta_i < x < \gamma_i, \quad i = 0, 1, \ldots, g, \\ \\ 0, & \text{otherwise}, \end{cases} \tag{13}$$

where $\alpha_i, \beta_i, \gamma_i \in R$, and α_i, γ_i are the lower and upper bounds of triangular fuzzy numbers, respectively. Suppose $TERMSET = \{l_0 = none, l_1 = worse, l_2 = bad, l_3 = fair, l_4 = good, l_5 = very\ good, l_6 = excellent\}$. The linguistic terms and their corresponding triangular fuzzy numbers in this $TERMSET$ are shown in Table 1. In addition, the triangular fuzzy numbers for the linguistic terms are shown in Figure 1.

Given a linguistic term l_h which is expressed with the triangular fuzzy numbers $(\alpha_h, \beta_h, \gamma_h)$, then the utility value of it is defined as follows [29]:

$$\text{utility}(l_h) = \frac{\alpha_h + 2\beta_h + \gamma_h}{4}, h = 0, 1, \ldots, g. \tag{14}$$

Thus, given the attribute values a_{ij} which are expressed with linguistic terms for attribute C_j, then the following method is used to normalize a_{ij}:

$$b_{ij} = \frac{\text{utility}(a_{ij})}{\sum_{t=1}^{m} \text{utility}(a_{tj})},$$

$$i = 1, \ldots, m,$$

$$j = 1, \ldots, n. \tag{15}$$

4.1.4. Normalize Linguistic Variables

Definition 6. An interval term $= [t_\eta, t_\delta]$ is called an uncertain linguistic variable, if both ends are linguistic terms, where t_η and t_δ are the lower and upper limits and are both linguistic terms.

Definition 7. Suppose $\text{term}_1 = [t_{\eta_1}, t_{\delta_1}]$ and $\text{term}_2 = [t_{\eta_2}, t_{\delta_2}]$ are two uncertain linguistic variables, then the distance between term_1 and term_2 are defined as follows [29]:

$$\text{sep}(\text{term}_1, \text{term}_2) = \frac{1}{2}\left(|\eta_1 - \eta_2| + |\delta_1 - \delta_2|\right). \tag{16}$$

Definition 8. Given the attribute C_j that is assessed by means of uncertain linguistic variables (e.g., $a_{ij} = [a_{ij}^L, a_{ij}^U]$), then the positive attribute value for it is defined as:

$$L\_\text{pos}_j^+ = \left[L\_\text{pos}_j^{+L}, L\_\text{pos}_j^{+U}\right], \tag{17}$$

where

$$L\_\text{pos}_j^{+L} = \max_{1 \le i \le m} \left\{a_{ij}^L\right\}, j = 1, \ldots, n. \tag{18a}$$

$$L\_\text{pos}_j^{+U} = \max_{1 \le i \le m} \left\{a_{ij}^U\right\}, j = 1, \ldots, n. \tag{18b}$$

Definition 9. Given the attribute C_j that is assessed by means of uncertain linguistic variables $a_{ij} = [a_{ij}^L, a_{ij}^U]$, then the grey correlation degree between $a_{ij} = [a_{ij}^L, a_{ij}^U]$ and $L\_pos_j^+$ is defined as follows:

TABLE 1: Linguistic terms and their corresponding triangular fuzzy numbers.

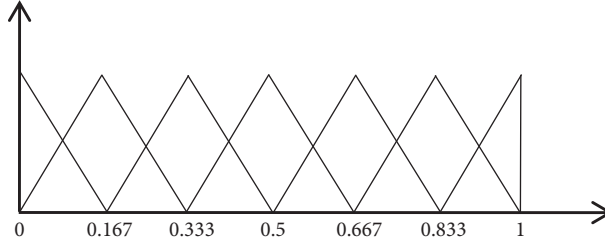| Level of linguistic terms	Linguistic terms	Triangular fuzzy numbers
l_0	None	(0, 0, 0.167)
l_1	Worse	(0, 0.167, 0.333)
l_2	Bad	(0.167, 0.333, 0.5)
l_3	Fair	(0.333, 0.5, 0.667)
l_4	Good	(0.5, 0.667, 0.833)
l_5	Very good	(0.667, 0.833, 1)
l_6	Excellent	(0.833, 1, 1)

FIGURE 1: Triangular membership functions of the linguistic terms.

$$b_{ij} = \frac{\min_i \min_j \left\{ \text{sep}\left(a_{ij}, L\_\text{pos}_j^+\right) \right\} + \rho \max_i \max_j \left\{ \text{sep}\left(a_{ij}, L\_\text{pos}_j^+\right) \right\}}{\text{sep}\left(a_{ij}, L\_\text{pos}_j^+\right) + \rho \max_i \max_j \left\{ \text{sep}\left(a_{ij}, L\_\text{pos}_j^+\right) \right\}}, i = 1, \ldots, m, j = 1, \ldots, n, \tag{19}$$

where *sep* () is the distance function between two uncertain linguistic variables given in Definition 7 and $L\_\text{pos}_j^+$ is the positive attribute value for attribute C_j as given in Definition 8. ρ is a parameter and is set as 0.5 usually.

Therefore, by means of the above steps (i.e., (16)–(19)), attribute values which are expressed with uncertain linguistic variables are normalized into the utility values between 0 and 1.

Based on the methods stated above, the heterogeneous attribute values in $A^k = [a_{ij}^k]_{m \times n}$ from an expert e^k are normalized, and denote the results as $B^k = [b_{ij}^k]_{m \times n}, k = 1, \ldots, P$.

4.2. Set Up Optimization Models to Determine Attribute Weights and Expert Weights. The following notations are adopted to facilitate describing the proposed approach.

Let $W = (w_1, w_2, \ldots w_n)^T$ denote the vector of attribute weights, where w_j is the weight of attribute C_j and w_j is between 0 and 1, $j = 1, \ldots, n$, while $\sum_{j=1}^n w_j = 1$.

Let $W^k = (w_1^k, w_2^k, \ldots, w_n^k)^T$ denote the vector of attribute weights that is derived from the preference information of expert e^k (i.e., $\widetilde{A}^k = [\widetilde{a}_{ij}^k]_{m \times n}$ and $R^k, k = 1, 2, \ldots, P$, where w_j^k is the weight of attribute C_j and w_j^k is between 0 and 1 $j = 1, \ldots, n, k = 1, \ldots, P$, while $\sum_{j=1}^n w_j^k = 1$.

By the method in (1), suppose $R^k = (r_1^k, r_2^k, \ldots, r_m^k)$ is normalized into $Y^k = (y_1^k, y_2^k, \ldots, y_m^k), k = 1, \&, P,$.

Let $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_P)^T$ denote the vector of expert weights, where λ_k is the weight of expert $e^k, k = 1, \ldots, P$.

4.2.1. Set Up the Optimization Model to Determine Attribute Weights

Definition 10. Given the normalized decision matrix $B^k = [b_{ij}^k]_{m \times n}$ from expert e^k and his/her overall preference on the alternatives $y_1^k, \ldots y_m^k$, the distance between the overall values of S_i and the expert's preference y_i^k on it is defined as follows:

$$\text{dis}_i^k = \left(\sum_{j=1}^n w_j^k b_{ij}^k - y_i^k \right)^2, i = 1, \ldots, m, k = 1, \ldots, P. \tag{20}$$

In order to determine the attribute weights that minimize the distance dis_i^k ($i = 1, \ldots, m$, $k = 1, \ldots, P$) as in Definition 10, the following model is set up:

$$\min \left(\sum_{j=1}^n w_j^k b_{ij}^k - y_i^k \right)^2, i = 1, \ldots, m, \tag{21a}$$

s.t.

$$\sum_{j=1}^n w_j^k = 1. \tag{21b}$$

$$0 \le w_j^k \le 1, j = 1, \ldots, n. \tag{21c}$$

After solving model (21a)–(21c), the optimal attribute weights can be obtained, as well as the values of the distance dis_i^k ($i = 1, \ldots, m$, $k = 1, \ldots, P$), which are denoted as the elements in the following matrix dis:

$$dis = \begin{pmatrix} dis_1^1 & \cdots & dis_1^p \\ \vdots & \ddots & \vdots \\ dis_m^1 & \cdots & dis_m^p \end{pmatrix}. \tag{22}$$

4.2.2. Set Up the Optimization Model to Determine Expert Weights. In order to determine the expert weights which minimize the elements in matrix dis (i.e., the values of dis_i^k, $i = 1, \ldots, m$, $k = 1, \ldots, P$) for all the alternatives, the function $\sum_{k=1}^{P} \bar{\lambda}_k dis_i^k$ should be minimized. The optimization goal is desired to be 0. However, the value of $\sum_{k=1}^{P} \bar{\lambda}_k dis_i^k$ is larger than 0, thus, the desirable goal is chosen as 0.0001 and the following optimization model is set up:

$$\min \sum_{i=1}^{m} \left(\sum_{k=1}^{P} \bar{\lambda}_k dis_i^k - 0.0001 \right)^2, \tag{23a}$$

s.t.

$$\sum_{k=1}^{P} \bar{\lambda}_k = 1, \tag{23b}$$

$$0 \le \bar{\lambda}_k \le 1, k = 1, \ldots, P. \tag{23c}$$

4.3. Set Up Neural Networks to Determine Attribute Weights and Expert Weights

4.3.1. Set Up a Neural Network to Determine Attribute Weights

(1) Network structure. According to models (21a)–(21c), a linear neural network is employed to train the attribute weights, as shown in Figure 2. The proposed neural network is composed of three components: the input layer, the output layer, and the expected output.

For the input layer, the attribute values of every alternative are used as the input data, for example, $(b_{i1}^k, b_{i2}^k, \ldots \ldots b_{in}^k)$. The number of attributes is used as the nodes of input layer, for example, n. Since there are m alternatives, then m sample data are adopted to train the neural network.

For the output layer, there is only one node and it denotes the output of the neural network. The output of the neural network is the weighted sum of the input data, while the connections between the input points and the output point are labeled with the attribute weights w_j^k.

The distance between the output of the neural network and the expected output y_i^k is used for correcting the attribute weights.
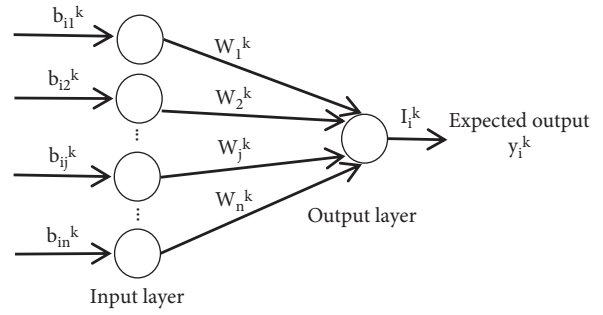

FIGURE 2: The structure of the proposed neural network for training the attribute weights.

(2) Reasoning Process. For alternative S_i, there are n attribute values that are the inputs of the neural network as proposed in Figure 2, and the corresponding output is as follows:

$$\text{net}_i^k = \sum_{j=1}^{n} b_{ij}^k w_j^k, i = 1, \ldots, m, k = 1, \ldots, p. \tag{24}$$

Since y_i^k is the expected output for the network, then the error function is defined as follows:

$$\text{Error}_i^k = \frac{1}{2} \left(\sum_{j=1}^{n} b_{ij}^k w_j^k - y_i^k \right)^2, i = 1, \ldots, m, k = 1, \ldots, P. \tag{25}$$

Thus, for all the alternatives, the overall error of the sample data is as follows:

$$\text{Error}^k = \frac{1}{2} \sum_{i=1}^{m} \left(\sum_{j=1}^{n} b_{ij}^k w_j^k - y_i^k \right)^2. \tag{26}$$

The training process is to make *Errork* as minimal as possible by adjusting the attribute weights *wik*. The weight coefficient correction (i.e., gradient descent) formula is as follows:

$$\Delta w_j^k = -\eta \sum_{i=1}^{m} b_{ij}^k \left(\sum_{j=1}^{n} b_{ij}^k w_j^k - y_i^k \right), j = 1, \ldots, n, k = 1, \ldots, P, \tag{27}$$

where η is the learning rate.

Based on the discussion above, the reasoning process is composed of the following steps:

Step 1. Initialize the proposed neural network as shown in Figure 2, and set the initial attribute weights as $w_j^k = [1/n, t1/nn, q \ldots h_{,1/n]} 1 \times n$.

Step 2. Add input sample and expected output. In order to ensure that the weight value adjustment is greater than 0 in the training process, the expected output of each sample is adjusted at will, and a constant is added at the same time. The new expected output is set as $y_i^k + \min 1/n \sum_{j=1}^{n} b_{ij}^k$.

Step 3. Calculate the actual output of the output layer according to formula (24).

Step 4. Calculate the error between the expected output value and the actual output according to formula (26); if the error meets the predefined training accuracy requirements, then stop the network training process, otherwise, go to step 5.

Step 5. Adjust the weighting coefficients between the output layer and the input layer according to (27).

Step 6. When the maximum number of training times is reached and the weights after training are positive, the training exit, otherwise go to step 2.
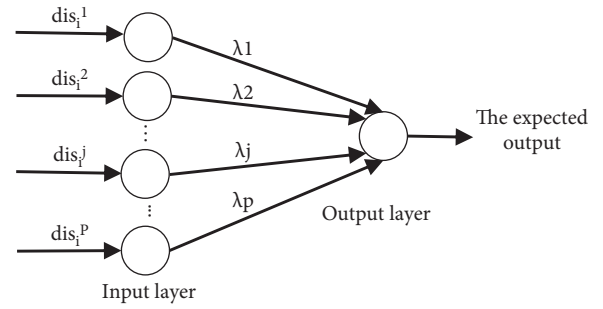
4.3.2. Set Up a Neural Network to Determine Expert Weights.
In order to fully reflect the roles of experts in the process of group decision-making, a neural network is proposed for training expert weights, as shown in Figure 3. This article continues to take similar steps to train and obtain the weights of experts, as stated earlier.

After training the neural network (as shown in Figure 2) by means of the sample data, that is, the attribute values of alternatives given by the P experts, the errors or distances between the expected output values and the actual outputs are obtained, as stated in (25). Then $di\,s_i^1$, $di\,s_i^2$ ..., $di\,s_i^P$, $i = 1,...,m$, are employed as the input for training the expert weights.

By denoting the training results as $\overline{\lambda}_k$, $k = 1,...,P$, the final expert weights can be obtained by means of the following formula:

$$\lambda_k = \frac{\overline{\lambda}_k}{\sum_{k=1}^{P} \overline{\lambda}_k}, k = 1, \ldots, P. \qquad (28)$$

Therefore, the attribute weights can be obtained as

$$\overline{w}_j = \sum_{k=1}^{P} \lambda_k w_j^k, j = 1, \ldots, n. \qquad (29)$$

4.4. Calculate the Overall Values of the Alternatives.
Based on the normalized decision matrix $B = [b_{ij}]_{m \times n}$, the attribute weights as in (29), and the expert weights in (28), the overall values of the alternatives can be obtained as

$$d_i = \sum_{k=1}^{P} \sum_{j=1}^{n} \lambda_k b_{ij}^k \overline{w}_j, i = 1, \ldots, m. \qquad (30)$$

The alternatives can be ranked according to their overall values as obtained in (30).

5. Illustrations and Comparisons

5.1. Illustrations.
In this illustration, five projects (i.e., alternatives, S_1, S_2, S_3, S_4, and S_5) are evaluated and chosen for investments. Four attributes are employed for the assessment task, that is, industry status (C_1), R&D investment (C_2), technological innovation (C_3), and market prospect (C_4), and they are assumed to be beneficial. Three experts are invited to evaluate the alternatives against the attributes and

FIGURE 3: The neural network for training expert weights.

give their overall evaluation on the alternatives are shown in Tables 2–4, respectively.

For the first attribute "industry status," the experts give their evaluation information by means of preference orderings. For the attribute "R&D investment," the experts give their evaluation information by means of intervals. For the attribute "Technological innovation," the experts give their evaluation information by means of linguistic terms from {none, worse, bad, fair, good, very good, excellent}. For the attribute "Market prospect," the experts give their evaluation information by means of uncertain linguistic variables, while $t_0 = $ none, $t_1 = $ worse, $t_2 = $ bad, $t_3 = $ fair, $t_4 = $ good, $t_5 = $ very good, $t_6 = $ excellent.

Based on the discussions above, the decision information from experts is normalized and the results are as follows:

$$B_1 = \begin{pmatrix} 0.5000 & 0.7143 & 0.3333 & 0.3333 \\ 0.3333 & 1.0000 & 0.6667 & 0.7500 \\ 0.6667 & 0.3333 & 1.0000 & 1.0000 \\ 0.5000 & 0.7143 & 0.4000 & 0.3333 \\ 1.0000 & 1.0000 & 0.4000 & 0.7500 \end{pmatrix},$$

$$B_2 = \begin{pmatrix} 0.3075 & 0.2808 & 0.3333 & 0.3333 \\ 0.7500 & 0.0000 & 0.6667 & 0.7500 \\ 0.0000 & 1.0000 & 1.0000 & 1.0000 \\ 1.0000 & 0.0000 & 0.4000 & 0.3333 \\ 0.6250 & 0.6526 & 0.4000 & 0.7500 \end{pmatrix}, \qquad (31)$$

$$B_3 = \begin{pmatrix} 0.4444 & 0.2344 & 0.0000 & 0.6548 \\ 1.0000 & 0.7504 & 1.0000 & 1.0000 \\ 0.7227 & 1.0000 & 0.4402 & 0.3652 \\ 0.0000 & 0.5000 & 0.5333 & 0.3333 \\ 0.0560 & 0.0000 & 0.4772 & 0.4223 \end{pmatrix}.$$

Expert e^1 gives his/her overall preference orderings on alternatives as (5,2,3,4,1), which can be normalized into $y_i^1 = \begin{bmatrix} 0 & 0.75 & 0.5 & 0.25 & 1 \end{bmatrix}$.

Expert e^2 gives his/her overall preference orderings on alternatives as (5,3,1,4,2), which can be normalized into $y_i^2 = \begin{bmatrix} 0 & 0.5 & 1 & 0.25 & 0.75 \end{bmatrix}$.

TABLE 2: Decision information from expert e^1 and overall preference orderings on alternatives.

	C_1 (industry status)	C_2 (R&D investment)	C_3 (technological innovation)	C_4 (market prospect)	R^1
S_1	3	[3.5, 4.0]	Bad	$[t_1, t_2]$	4
S_2	1	[3.8, 4.1]	Very good	$[t_2, t_4]$	2
S_3	2	[3.0, 3.5]	Excellent	$[t_2, t_5]$	1
S_4	5	[2.8, 3.5]	Bad	$[t_1, t_3]$	5
S_5	4	[3.2, 3.6]	Good	$[t_1, t_3]$	3

TABLE 3: Decision information from expert e^2 and overall preference orderings on alternatives.

	C_1 (industry status)	C_2 (R&D investment)	C_3 (technological innovation)	C_4 (market prospect)	R^2
S_1	4	[2.8, 3.2]	Bad	$[t_1, t_2]$	5
S_2	3	[3.5, 4.0]	Excellent	$[t_2, t_3]$	2
S_3	1	[3.3, 3.5]	Fair	$[t_2, t_4]$	3
S_4	5	[3.0, 3.5]	Fair	$[t_1, t_2]$	4
S_5	2	[4.0, 4.5]	Very good	$[t_2, t_3]$	1

TABLE 4: Decision information from expert e^3 and overall preference orderings on alternatives.

	C_1 (industry status)	C_2 (R&D investment)	C_3 (technological innovation)	C_4 (market prospect)	R^3
S_1	4	[2.8, 3.2]	Bad	$[t_1, t_2]$	5
S_2	2	[3.2, 4.2]	Fair	$[t_3, t_4]$	3
S_3	1	[4.2, 4.5]	Good	$[t_3, t_5]$	1
S_4	5	[3.0, 3.8]	Fair	$[t_3, t_4]$	4
S_5	3	[3.5, 4.0]	Very good	$[t_3, t_5]$	2

Expert e^3 gives his/her overall preference orderings on alternatives as (4, 1, 2, 3, 5), which can be normalized into $y_i^3 = \begin{bmatrix} 0.25 & 1 & 0.75 & 0.5 & 0 \end{bmatrix}$.

Based on Figure 2, the following neural network is designed for simulations by means of MatLab. As stated in Figure 4, there are four input nodes, which correspond to the four attributes, and five samples correspond to the attribute values of the five alternatives.

The initial attribute weights are set as $w^k = (0.25, 0.25, 0.25, 0.25)$ for the three experts, $k = 1,2,3$. The learning rate of the network is set as 0.001 and the maximum number of iterations is set as 3000.

Based on the normalized decision information of expert e^1, after going through 3000 times of iterations, the attribute weights and errors are obtained and stated in Tables 5 and 6, respectively.

Based on the normalized decision information of expert e^2, after going through 3000 times of iterations, the attribute weights and errors are obtained and stated in Tables 7 and 8, respectively.

Based on the normalized decision information of expert e^3, after going through 3000 times of iterations, the attribute weights and errors are obtained and stated in Tables 9 and 10, respectively.

Furthermore, in order to determine the expert weights, based on the neural network in Figure 3, the neural network for training expert weights is applied in MatLab is stated in Figure 5.

The number of input neurons is three and the number of output neurons is one. The inputs are the values in the
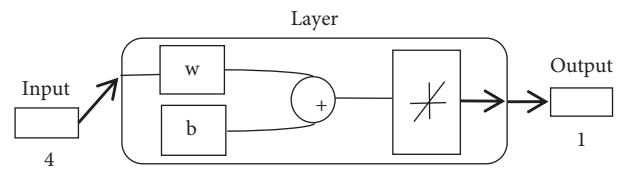

FIGURE 4: The neural network in MatLab for the simulations on training attribute weights.

matrix dis obtained by the above process for training the attribute weights, stated as follows:

$$dis = \begin{pmatrix} 0.0368 & 0.0057 & 0.1893 \\ 0.4913 & 0.3196 & 0.3393 \\ 0.2841 & 0.4971 & 0.2994 \\ 0.2055 & 0.2187 & 0.3385 \\ 0.6133 & 0.4604 & .0.0257 \end{pmatrix}. \quad (32)$$

The expected output is set as 0.0001. The learning rate of the network is set as 0.001 and the maximum number of iterations is set as 3000. After training the above neural network in Figure 5, the expert weights are obtained, as stated in Table 11.

Therefore, the attribute weights can be obtained as $W = (0.1839, 0.3492, 0.1927, 0.2742)$. The overall values of the alternatives can be obtained as $d_1 = 0.3818$, $d_2 = 0.6963$, $d_3 = 0.72$, $d_4 = 0.4028$, and $d_5 = 0.5489$. Thus, the final rankings of the alternatives is $s_3 \succ s_2 \succ s_5 \succ s_4 \succ s_1$.

TABLE 5: Attribute weights after training the neural network for the decision information of expert e^1.

	w_1^1	w_2^1	w_3^1	w_4^1
Attribute weights	0.2384	0.3516	0.1159	0.2941

TABLE 6: Errors after training the neural network for the decision information of expert e^1.

Samples	S_1	S_2	S_3	S_4	S_5
Errors	0.0368	0.4913	0.2841	0.2055	0.6133

TABLE 7: Attribute weights after training the neural network for the decision information of expert e^2.

	w_1^2	w_2^2	w_3^2	w_4^2
Attribute weights	0.1665	0.2931	0.2297	0.3107

TABLE 8: Errors after training the neural network for the decision information of expert e^2.

Samples	S_1	S_2	S_3	S_4	S_5
Errors	0.0057	0.3196	0.4971	0.2187	0.4604

TABLE 9: Attribute weights after training the neural network for the decision information of expert e^3.

	w_1^3	w_2^3	w_3^3	w_4^3
Attribute weights	0.1428	0.4023	0.2382	0.2167

TABLE 10: Errors after training the neural network for the decision information of expert e^3.

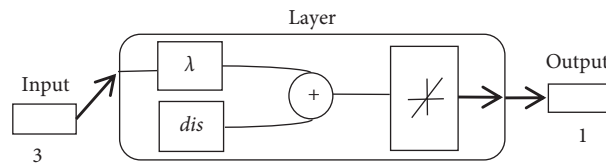| Samples	S_1	S_2	S_3	S_4	S_5
Errors	0.1893	0.3393	0.2994	0.3385	0.0257

FIGURE 5: The neural network in MatLab for the simulations on training expert weights.

TABLE 11: Expert weights obtained after training neural network in Figure 5.

Experts	e^1	e^2	e^3
Expert weights	0.3500	0.3265	0.3235

5.2. Comparisons with Other Methods. In order to display the advantage of the neural network-based approach proposed in this article, a comparison is made with the maximum deviation method with respect to the three experts' decision information. As stated in Tables 12–14, the results of the attribute weights are different. The reason is that the maximum deviation method deals with the attribute values in the decision matrix. However, the neural network-based approach proposed in this article integrates the attribute values in the decision matrix with the experts' preference

TABLE 12: Comparison based on expert e^1's decision information.

Methods	Attribute weights	Consider attribute values?	Consider experts' preference Information?	Both attribute values and experts' preference	Overall values of alternatives
Maximum deviation method	$w_1 = 0.2319$ $w_2 = 0.2503$ $w_3 = 0.2473$ $w_4 = 0.2705$	consider	Not consider	ONLY consider attribute values	Need calculate
Neural network approach	$w_1 = 0.2384$ $w_2 = 0.3516$ $w_3 = 0.1159$ $w_4 = 0.2941$	consider	consider	Both	Neural network outputs

TABLE 13: Comparison based on expert e^2's decision information.

Methods	Attribute weights	Consider attribute values?	Consider experts' preference Information?	Both attribute values and experts' preference	Overall values of alternatives
Maximum deviation method	$w_1 = 0.2835$ $w_2 = 0.3166$ $w_3 = 0.1910$ $w_4 = 0.2089$	consider	Not consider	ONLY consider attribute values	Need calculate
Neural network approach	$w_1 = 0.1665$ $w_2 = 0.2931$ $w_3 = 0.2297$ $w_4 = 0.3107$	consider	consider	Both	Neural network outputs

TABLE 14: Comparison based on expert e^3's decision information.

Methods	Attribute weights	Consider attribute values?	Consider experts' preference Information?	Both attribute values and experts' preference	Overall values of alternatives
Maximum deviation method	$w_1 = 0.2997$ $w_2 = 0.2827$ $w_3 = 0.2352$ $w_4 = 0.1824$	consider	Not consider	ONLY consider attribute values	Need calculate
Neural network approach	$w_1 = 0.1428$ $w_2 = 0.4023$ $w_3 = 0.2382$ $w_4 = 0.2167$	consider	consider	Both	Neural network outputs

information on alternatives, by means of training the neural networks and memorizing the two sources of information by weights.

Furthermore, in terms of calculating the overall values of alternatives, for the new data of attribute values, the maximum deviation method has to be applied to figure out the attribute weights and calculate the overall values of alternatives. However, with the new data of attribute values, the neural network-based approach proposed in this article will obtain the overall values of alternatives without solving the optimization models to get attribute eights again, since the neural network will output the overall values of alternatives.

6. Conclusions

By means of setting up two optimization models and designing two neural networks, an approach is proposed to deal with the multi-attribute group decision-making problems with heterogeneous preference information on

attribute values and overall preference orderings on alternatives. The first neural network is trained for attribute weights by using attribute values as the inputs and overall preference orderings on alternatives as the expected outputs. The second neural network is trained for deriving the expert weights.

The merits of the proposed approach lie in three aspects: (1) setting two models to optimize the attribute weights and the expert weights; (2) based on the optimization model for attribute weights, the neural network is designed and trained, which remembers experts' preference against attribute values and on alternatives. The trained neural network can be used for new entries of preference information on attribute values and obtain the overall values of alternatives automatically, which satisfies the experts' preference on alternatives; (3) based on the optimization model for expert weights, the neural network is designed and trained for memorizing the experts' weights, which would be the guidance for decision-making process.

To summarize, the contributions of this article are as follows: (1) the proposed approach presents a new way of solving the uncertain multi-attribute group decision-making problems with heterogeneous preference information on attribute values and overall preference orderings on alternatives. (2) The proposed neural networks enable the experts' preference information to be remembered and allow new data entry on attribute values, as well as obtain the overall values of alternatives without solving the optimization models again. The proposed approach and the neural networks provide guidance for decision-makers in uncertain multiple attribute group decision-making problems with preference information on alternatives.

In group decision-making under uncertainty, it is important to reach consensus. The hot studies will focus on consensus reaching process in MAGDM with heterogeneous preference information [30–32]. The following studies will be consensus reaching for personalized individual semantics-based social network group decision-making by means of neural networks. In addition, personalized individual semantics-based consistency control and consensus reaching in group decision-making will also be the hot topic [30–32].

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare no conflicts of interest.

References

[1] 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.

[2] Y. H. Lin, P. C. H. Lee, T. P. Chang, and H. I. Ting, "Multi-attribute group decision making model under the condition of uncertain information," *Automation in Construction*, vol. 17, no. 6, pp. 792–797, 2008.

[3] T. L. Saaty, "Decision Making with the Analytic Hierarchy Process," *International Journal of Services Sciences*, vol. 1, no. 1, pp. 83–98, 2008.

[4] P. Gupta, M. K. Mehlawat, N. Grover, and W. Pedrycz, "Multi-attribute group decision making based on extended TOPSIS method under interval-valued intuitionistic fuzzy environment," *Applied Soft Computing*, vol. 69, pp. 554–567, 2018.

[5] S. Abootalebi, A. Hadi-Vencheh, and A. Jamshidi, "An improvement to determining expert weights in group multiple attribute decision making problem," *Group Decision and Negotiation*, vol. 27, no. 2, pp. 215–221, 2018.

[6] Y. C. Dong, Y. T. Liu, H. M. Liang, F. Chiclana, and E. Herrera-Viedma, "Strategic weight manipulation in multiple attribute decision making," *Omega*, vol. 75, pp. 154–164, 2018.

[7] Y. T. Liu, Y. C. Dong, H. M. Liang, F. Chiclana, and E. Herrera-Viedma, "Multiple Attribute Strategic Weight Manipulation With Minimum Cost in a Group Decision Making Context With Interval Attribute Weights Information," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 49, pp. 1981–1992, 2019.

[8] Z. P. Fan, J. Ma, and Q. Zhang, "An approach to multiple attribute decision making based on fuzzy preference information on alternatives," *Fuzzy Sets and Systems*, vol. 131, no. 1, pp. 101–106, 2002.

[9] Y. M. Wang and C. Parkan, "Multiple attribute decision making based on fuzzy preference information on alternatives: Ranking and weighting," *Fuzzy Sets and Systems*, vol. 153, no. 3, pp. 331–346, 2005.

[10] H. Deng, C. H. Yeh, and R. J. Willis, "Inter-company comparison using modified TOPSIS with objective weights," *Computers & Operations Research*, vol. 27, no. 10, pp. 963–973, 2000.

[11] X. Xu, "A note on the subjective and objective integrated approach to determine attribute weights," *European Journal of Operational Research*, vol. 156, no. 2, pp. 530–532, 2004.

[12] Z. Wu and Y. Chen, "The maximizing deviation method for group multiple attribute decision making under linguistic environment," *Fuzzy Sets and Systems*, vol. 158, no. 14, pp. 1608–1617, 2007.

[13] G. W. Wei, "Maximizing deviation method for multiple attribute decision making in intuitionistic fuzzy setting," *Knowledge-Based Systems*, vol. 21, no. 8, pp. 833–836, 2008.

[14] Y. M. Wang and C. Parkan, "A general multiple attribute decision-making approach for integrating subjective preferences and objective information," *Fuzzy Sets and Systems*, vol. 157, no. 10, pp. 1333–1345, 2006.

[15] H. M. Liang, Q. Gong, Y. C. Dong, and Z. G. Ding, "A consistency-based approach to multiple attribute decision making with preference information on alternatives," *Computers & Industrial Engineering*, vol. 119, pp. 360–369, 2018.

[16] C. Yue, "Projection-based approach to group decision-making with hybrid information representations and application to software quality evaluation," *Computers & Industrial Engineering*, vol. 132, pp. 98–113, 2019.

[17] V. C. S. Lee and H. Tshung Wong, "A multivariate neuro-fuzzy system for foreign currency risk management decision making," *Neurocomputing*, vol. 70, no. 4–6, pp. 942–951, 2007.

[18] T. Efendigil, S. Önüt, and C. Kahraman, "A decision support system for demand forecasting with artificial neural networks and neuro-fuzzy models: A comparative analysis," *Expert Systems with Applications*, vol. 36, no. 3, pp. 6697–6707, 2009.

[19] S. van Cranenburgh and A. Alwosheel, "An artificial neural network based approach to investigate travellers' decision rules," *Transportation Research Part C: Emerging Technologies*, vol. 98, pp. 152–166, 2019.

[20] W. B. Ren, K. Z. Wu, Q. S. Gu, and Y. G. Hu, "Intelligent decision making for service providers selection in maintenance service network: An adaptive fuzzy-neuro approach," *Knowledge-Based Systems*, vol. 190, Article ID 105263, 2020.

[21] D. Çelebi and D. Bayraktar, "An integrated neural network and data envelopment analysis for supplier evaluation under incomplete information," *Expert Systems with Applications*, vol. 35, no. 4, pp. 1698–1710, 2008.

[22] D. Golmohammadi, "Neural network application for fuzzy multi-criteria decision making problems," *International Journal of Production Economics*, vol. 131, no. 2, pp. 490–504, 2011.

[23] Y. T. Liu, H. J. Zhang, Y. Z. Wu, and Y. C. Dong, "Ranking range based approach to MADM under incomplete context and its application in venture investment evaluation,"

Technological and Economic Development of Economy, vol. 25, no. 5, pp. 877–899, 2019.

[24] H. J. Zhang, Y. C. Dong, J. Xiao, F. Chiclana, and E. Herrera-Viedma, "Personalized individual semantics-based approach for linguistic failure modes and effects analysis with incomplete preference information," *IISE Transactions*, vol. 52, no. 11, pp. 1275–1296, 2020.

[25] J. Xiao, X. L. Wang, and H. J. Zhang, "Exploring the Ordinal Classifications of Failure Modes in the Reliability Management: An Optimization-Based Consensus Model with Bounded Confidences," *Group Decision and Negotiation*, vol. 31, no. 1, pp. 49–80, 2022.

[26] J. Lin and R. Q. Chen, "A Novel Group Decision Making Method Under Uncertain Multiplicative Linguistic Environment for Information System Selection," *IEEE Access*, vol. 7, Article ID 19848, 2019.

[27] L. H. Yang and B. L. Li, "Multiple-Valued Neutrosophic Uncertain Linguistic Sets With Dombi Normalized Weighted Bonferroni Mean Operator and Their Applications in Multiple Attribute Decision Making Problem," *IEEE Access*, vol. 8, pp. 5906–5927, 2020.

[28] F. Herrera, L. Martınez, and P. J. Sánchez, "Managing non-homogeneous information in group decision making," *European Journal of Operational Research*, vol. 166, no. 1, pp. 115–132, 2005.

[29] Z. S. Xu, "Induced uncertain linguistic OWA operators applied to group decision making," *Information Fusion*, vol. 7, no. 2, pp. 231–238, 2006.

[30] J. Wu, Q. Hong, M. S. Cao, Y. J. Liu, and H. Fujita, "A group consensus-based travel destination evaluation method with online reviews," *Applied Intelligence*, vol. 52, no. 2, pp. 1306–1324, 2022.

[31] M. S. Cao, Y. J. Liu, T. T. Gai, M. Zhou, H. Fujita, and J. Wu, "A Comprehensive Star Rating Approach for Cruise Ships Based on Interactive Group Decision Making with Personalized Individual Semantics," *Journal of Marine Science and Engineering*, vol. 10, no. 5, p. 638, 2022.

[32] Y. Gao and Z. Zhang, "Consensus reaching with non-cooperative behavior management for personalized individual semantics-based social network group decision making," *Journal of the Operational Research Society*, vol. 1, p. 18. in press, 2021.