Soft Comput (2006) 10: 543–554
DOI 10.1007/s00500-005-0484-x

ORIGINAL PAPER

J.-R. Chang · T.-H. Ho · C.-H. Cheng · A.-P. Chen

Dynamic fuzzy OWA model for group multiple criteria decision making

Published online: 18 May 2005

Abstract Obtaining relative weights in MCDM problems is a very important issue. The Ordered Weighted Averaging (OWA) aggregation operators have been extensively adopted to assign the relative weights of numerous criteria. However, previous aggregation operators (including OWA) are independent of aggregation situations. To solve the problem, this study proposes a new aggregation model – dynamic fuzzy OWA based on situation model, which can modify the associated dynamic weight based on the aggregation situation and can work like a "magnifying lens" to enlarge the most important attribute dependent on minimal information, or can obtain equal attribute weights based on maximal information. Two examples are adopted in this paper for comparison and showing the effects under different weights.

Keywords Fuzzy multiple criteria decision making (FMCDM) · Ordered weighted averaging (OWA) · Dynamic fuzzy OWA model · Linguistic variable

J.-R. Chang (✉) · C.-H. Cheng
Department of Information Managements,
National Yunlin University of Science and Technology,
123 University Road, section 3, Touliu, Yunlin 640, Taiwan
Tel.: +886-5-5342601
Fax: +886-5-5312077
E-mail: g9120806@yuntech.edu.tw

J.-R. Chang
Graduate Institute of Management,
National Yunlin University of Science and Technology,
123 University Road, section 3, Touliu, Yunlin 640, Taiwan

T.-H. Ho · A.-P. Chen
Institute of Information Management,
National Chiao Tung University,
1001 Ta Hsueh Road, Hsinchu 300, Taiwan

T.-H. Ho
Department of Information Management,
Nan Jeon Institute of Technology,
178, Zhao-Qin Road, Yenshui, Tainan, Taiwan

1 Introduction

Information aggregation can be applied to many situations, including neural networks, fuzzy logic controllers, expert systems, and multi-criteria decision support systems [14]. In a vague condition, fuzzy set theory [30] can provide an attractive connection to represent uncertain information and can aggregate them properly. The existing aggregation operators are, in general, the t-norm [26], t-conorm [26], mean operators [11], Yager's operator [28] and γ-operator [32].

Multi-criteria decision making (MCDM) models are characterized to evaluate a finite set of alternatives. The main purpose of solving MCDM problems is to measure the overall preference values of the alternatives. Two reasons reveal the importance of obtaining relative weights in MCDM problems. First, numbers of approaches have been proposed to assess criteria weights, which are then used explicitly to aggregate specific priority scores [5–7, 17, 19, 23, 24, 27, 31]. Second, some experiments [1, 22, 25] demonstrate that different approaches for deriving weights may lead to different results [16].

When an attempt is made to solve the MCDM problem by aggregating the information of each attribute in many disciplines, a problem of aggregating criteria functions to form overall decision functions occurs owing to these criteria always being interdependent. One extreme is the situation in which we hope that all the criteria will be satisfied ("and" situation), while another situation is the case in which satisfying simple criteria satisfaction is that any of the criteria is all we desire ("or" situation) [27, 29]. In 1988, Yager [27] first introduced the concept of OWA operators to solve this problem. The OWA operators have the ability to provide an aggregation lying between these two extremes, so it more fit the thought of human being (between the "and" and "or" situations) [27]. O'Hagan [18] is the first to use the concept of entropy in the OWA operation, but situation factor has not yet been taken into the consideration of this method. Mesiar and Saminger [15] have shown that in the class of OWA operators is on the domination over the t-norm, and the domination of OWA operators and related operators over continuous Archimedean t-norms is also discussed.

However, these aggregation operators [8,11,28,32] are independent of their situations and cannot reflect change in situations [4]. To resolve this problem, this study proposes a dynamic OWA aggregation model based on the faster OWA operator, which has been introduced by Fuller and Majlender [8] and can work like a magnifying lens and adjust its focus based on the sparest information to change the dynamic attribute weights to revise the weight of each attribute based on aggregation situation, and then to provide suggestions to decision maker (DM).

To verify the proposed model, two examples are adopted in this paper. The first example is to evaluate aggregative risks in software development with three software projects [13]. The results of proposed method will compare with Lee [13] and Chen [2]. The second example is to solve the distribution center selection problem in Taiwan [3], and the results will also compare with the Chen's method [3].

The rest of this paper is organized as follows. Section 2 presents a basic concept of the OWA operator. Section 3 introduces the proposed model and a generalized algorithm. The verification and comparison based on experimental results for two examples are introduced in Sect. 4. Subsequently, Sect. 5 discuss the finding and give some suggestions. Conclusions are finally made in Sect. 6.

2 OWA operator

The OWA operator [27,29] is an important aggregation operator within the class of weighted aggregation methods. Many related studies have been conducted in recent years. For example, Fuller and Majlender [8] have used Lagrange multipliers to derive a polynomial equation to solve constrained optimization problem and to determine the optimal weighting vector. Meanwhile, Smolikova and Wachowiak [23] have described and compared aggregation techniques for expert multi-criteria decision-making method. Furthermore, Ribeiro and Pereira [20] have presented an aggregation schema based on generalized mixture operators using weighting functions, and have compared it with these two standard aggregation method: weighting averaging and ordered weighted averaging in the context of multiple attribute decision making. The main concepts of this approach are derived from the OWA operators of Yager [27] and Fuller and Majlender [8]. This section introduces the main content of their methods.

2.1 Yager's OWA

According to previous studies, t-norm and t-conorm are based on the theory of logic [26], and the mean operators [11] are based on the mathematical properties of averaging. However, in the opinion of Choi [4], these types of aggregation operators are independent of aggregation situation. Even though Yager's operator [28] and γ-operator [32] are suggested as an aggregation method using parameter, at present, the definition of such a parameter is still missing [4].

Yager [27] proposed an OWA operator, which had the ability to get optimal weights of the attributes based on the rank of these weighting vectors after aggregation process (reference to Definition 1).

Definition 1. An OWA operator of dimension n is a mapping F: $R^n \to R$, that has an associated weighting vector $W = [w_1, w_2, \ldots\ldots, w_n]^T$ of having the properties

$$\sum_i w_i = 1, \quad \forall w_i \in [0,1], \quad i = 1, \ldots, n,$$

$$\text{and such that } f(a_{1,\ldots,}a_n) = \sum_{j=1}^{n} w_j\, b_j \tag{1}$$

where b_j is the jth largest element of the collection of the aggregated objects $\{a_{1,\ldots,}a_n\}$

Yager [27,29] also introduced two important characterizing measures in respect to the weighting vector W of an OWA operator. The first one was the measure of *orness* of the aggregation, which was defined as

$$\text{Orness}(W) = \frac{1}{n-1} \sum_{i=1}^{n} (n-i)\, w_i \tag{2}$$

And, the second one, implying the measure of dispersion of the aggregation, was defined as

$$\text{Disp}(W) = -\sum_{i=1}^{n} w_i \ln W_i\,. \tag{3}$$

And it measures the degree to which W takes into account all information in the aggregation.

O'Hagan [18] suggested a method which combines the principle of maximum entropy [9,10,12,21] and Yager's approach [27] to determine a special class of OWA operators having the maximal entropy of the OWA weights for a given level of *orness*. This approach was based on the solution of the following problem:

$$\text{Maximize the function } -\sum_{i=1}^{n} w_i \ln W_i$$

$$\text{Subject to the constraints } \alpha = \frac{1}{n-1} \sum_{i=1}^{n} (n-i)\, w_i\,,$$

$$0 \le \alpha \le 1$$

$$\sum_i w_i = 1, \quad \forall w_i \in [0,1], \quad i = 1, \ldots, n, \tag{4}$$

2.2 Fuller and Majlender's OWA

Fuller and Majlender [8] used the method of Lagrange multipliers to transfer Yager's OWA equation to a polynomial equation, which can determine the optimal weighting vector. By their method, the associated weighting vector is easily obtained by (5)–(7).

$$\ln w_j = \frac{j-1}{n-1}\ln w_n + \frac{n-j}{n-1}\ln w_1 \Rightarrow w_j = \sqrt[n-1]{w_1^{n-j}\, w_n^{j-1}}, \tag{5}$$

and

$$w_n = \frac{((n-1)\alpha - n)\, w_1 + 1}{(n-1)\alpha + 1 - n\, w_1}, \tag{6}$$

then

$$w_1\,[(n-1)\,\alpha + 1 - n\,w_1]^n = [(n-1)\,\alpha]^{n-1}\,[((n-1)\,\alpha - n)\,w_1 + 1]\,. \tag{7}$$

So the optimal value of w_1 should satisfy equation (7). When w_1 is computed, we can determine w_n from equation (6) and then the other weights are obtained from equation (5). In a special case, when $w_1 = w_2 = \cdots\cdots = w_n = 1/n \Rightarrow \text{disp}(W) = \ln n$ which is optimal solution to equation (5) for $\alpha = 0.5$.

3 New dynamic OWA aggregation model

3.1 Dynamic fuzzy OWA model

After comparing the operators in [4,8,11,20,23,27,28,32] with the OWA operator, the paper finds that the OWA operator has the rational aggregation result, and more closely fits the thoughts of human beings (between the "and" and "or" situations) [27]. Moreover, under the circumstances of maximal information entropy, the OWA operator can get the optimum result of the aggregation. However, it lacks the ability to reflect the aggregative situation during the aggregation process because the previous OWA operators use a common parameter (i.e. α), but do not view it as the situational factor. To maintain the useful character of the OWA (rational aggregation result) and correct the shortcomings (lack to reflect the aggregative situation), this study adds two main concepts (see Fig. 1):

(1) Facilitating dynamic aggregation result (attribute weights) by feedback.
(2) Changing the attribute weights based on situation (with situation parameter α).

After joining these two characters with the fundamental OWA aggregation model, this work proposes a new fuzzy OWA aggregation model, and clarifies the main differences between the proposed model and other aggregation methods in Table 1. This new model not only has the ability to modify forecasting results of functions corresponding with the aggregative situation, but also can obtain associated attribute weights that rely on the OWA operator matching the model of human thoughts.

The first concept of modifying the aggregation dynamic attribute weights is the process, which is given to experts who want to evaluate the projects different weights. In this way, the experts will have different affects on integral result after evaluation. For example, if the evaluative time is regarded as a criterion to measure the degree of information quality, the newly-coming experts will be assigned a higher weight. This step can enable the newly-coming experts to have more influence on the attribute weights and individual project evaluation. Consequently, different attribute ratings can obtain dissimilar attribute weights and also different final proposed solutions for reference by decision makers.

Second, the concept of changing weights of each attribute "based on situation" is that the decision maker (or project manager) determines what is the value of parameter α from information entropy of actual aggregative situation. Therefore, the proposed model can be used to obtain attribute weights by rating them after OWA aggregation according to α. The main advantage of this concept is that the model can be treated as a magnifying lens to determine the most important attribute (assign weight = 1) based on the sparest information (*i.e.* optimistic and $\alpha = 0$ or 1) situation. On the other hand, when $\alpha = 0.5$ (moderate situation), the proposed model can obtain attribute weights (equal weights of attributes) based on maximal information.

3.2 Algorithm for the proposed model

The steps of proposed algorithm are as follows:

Step 1 Build hierarchical structure model from determination problem and number (N) of attributes/criteria.
Step 2 Obtain opinions of domain experts and then collect their evaluative attribute weights of attributes in respect to the hierarchical structure model.
Step 3 List the feasible projects/alternatives, and request the experts to evaluate the grades of these projects.
Step 4 If no new expert is available, execute step 5. If the experts do not have significant orderings, assign equal weight for evaluation. Otherwise, perform the OWA aggregation process to obtain the weights of experts for evaluation.
Step 5 The weights of each expert multiply their evaluative attribute weights to form the aggregative weights of attributes.
Step 6 Sort the attribute weights and execute OWA aggregation (by equations (5)–(7)) to obtain refined attribute weights.
Step 7 If the aggregative weights of sub-attributes are exist, to distribute the refined weight(s) of the sub-attributes of each attribute based on the ratio of weights of these sub-attributes given by the experts.
Step 8 Multiply the weights of the attributes by their project grades, and then rank their orderings to make reference solution to the decision maker.

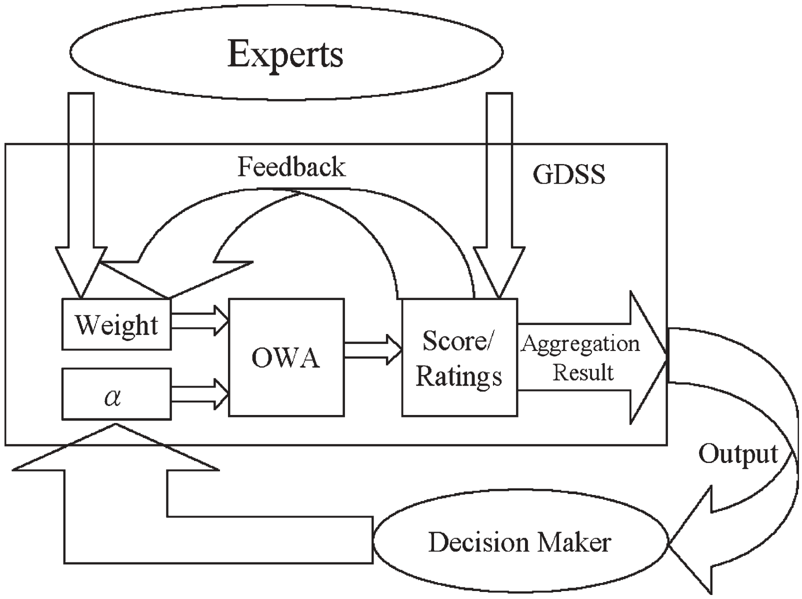

Fig. 1 A schematic view of proposed model

Table 1 Main differences between proposed model and other aggregation methods

	Yager's OWA [27]	Fuller and Majlender's OWA [8]	Choi's operator [4]	Lee's two algorithm [13]	Chen's algorithm [2]	Chen's [3] algorithm	Proposed dynamic OWA
Aggregation operator	Yes	Yes	Yes	No	No	No	Yes
Situation parameter (α)	Partial*	Partial*	Yes	No	No	No	Yes
Feedback	No	No	No	No	No	No	Yes
Fuzzy Input	No	No	No	Yes	Yes	Yes	Yes

*The OWA operators use a common parameter (i.e. α), but they do not view it as the situational factor.

4 Verification and comparison

In this section, two examples are adopted to illustrate and verify the proposed model: (1) evaluating the risks of software development; (2) external evaluation of distribution centers in logistic. The results and comparisons of each example are also described.

4.1 Example I: evaluation of software development risk

To compare the fuzzy group decision making methods, this section first introduces two algorithms developed by Lee for group decision making structure model of risk in software development, and then presents the algorithm of Chen. We also adjust the proposed algorithm in Sect. 4.1.3 to fit this example.

4.1.1 Lee's [13] algorithms

Lee [13] assumes that there is a group of n experts (D_1,D_2, ..., D_n) to assess the risks for a project in software development. Let the symbol $\tilde{W}_{2(j,h)}$ denote the relative importance weight given by expert D_j to attribute h, and let $\tilde{W}_{1(j,h,k)}$, $\tilde{r}_{(j,h,k)}$ and $\tilde{i}_{(j,h,k)}$denote the weight, the grade of risk, and the grade of importance given to the risk item X_{hk} for expert D_j's assessment data ($j = 1, 2, \ldots, n$; $h = 1, 2, \ldots, 6$; $k = 1, 2, \ldots, n(h)$, and $n(h)$ is the number of risk items for attributes X_h), and (+) and ($\times$) denote the addition and multiplication operators of triangular fuzzy numbers (TFNs), respectively. Table 2 shows an example of the contents of the hierarchical structure model, where

$$\tilde{W}_{2(j,h)} = (a_{2(j,h)}, b_{2(j,h)}, c_{2(j,h)}) \tag{8}$$

$$\tilde{W}_{1(j,h,k)} = (a_{1(j,h,k)}, b_{1(j,h,k)}, c_{1(j,h,k)}) \tag{9}$$

$$\tilde{r}_{(j,h,k)} = (a_{3(j,h,k)}, b_{3(j,h,k)}, c_{3(j,h,k)}) \tag{10}$$

$$\tilde{i}_{(j,h,k)} = (a_{4(j,h,k)}, b_{4(j,h,k)}, c_{4(j,h,k)})\,. \tag{11}$$

1. Lee's first algorithm (Algorithm I): This algorithm averages each parameter individually and then aggregates the results to produce a final rate of aggregative risk. The main context of the algorithm is as follows:
   i. Calculate the average of each parameter on the fuzzy data of n decision makers.
   ii. First-stage assessment. Establish a fuzzy assessment matrix $\tilde{M}(X_h)$ for each attribute X_h, and then use these fuzzy assessment matrices to evaluate the first-stage aggregative assessment matrix $\tilde{R}_{1(h)}$ for each attribute X_h.

Table 2 The contents of the hierarchical structure model for decision maker D_j[13]

Attribute	Risk item	Weight-2	Weight-1	Grade of risk	Grade of importance
X_1		$\tilde{W}_{2(j,1)}$			
	X_{11}		$\tilde{W}_{1(j,1,1)}$	$\tilde{r}_{(j,1,1)}$	$\tilde{i}_{(j,1,1)}$
X_2		$\tilde{W}_{2(j,2)}$			
	X_{21}		$\tilde{W}_{1(j,2,1)}$	$\tilde{r}_{(j,2,1)}$	$\tilde{i}_{(j,2,1)}$
	X_{22}		$\tilde{W}_{1(j,2,2)}$	$\tilde{r}_{(j,2,2)}$	$\tilde{i}_{(j,2,2)}$
	X_{23}		$\tilde{W}_{1(j,2,3)}$	$\tilde{r}_{(j,2,3)}$	$\tilde{i}_{(j,2,3)}$
	X_{24}		$\tilde{W}_{1(j,2,4)}$	$\tilde{r}_{(j,2,4)}$	$\tilde{i}_{(j,2,4)}$
X_3		$\tilde{W}_{2(j,3)}$			
	X_{31}		$\tilde{W}_{1(j,3,1)}$	$\tilde{r}_{(j,3,1)}$	$\tilde{i}_{(j,3,1)}$
	X_{32}		$\tilde{W}_{1(j,3,2)}$	$\tilde{r}_{(j,3,2)}$	$\tilde{i}_{(j,3,2)}$
X_4		$\tilde{W}_{2(j,4)}$			
	X_{41}		$\tilde{W}_{1(j,4,1)}$	$\tilde{r}_{(j,4,1)}$	$\tilde{i}_{(j,4,1)}$
	X_{42}		$\tilde{W}_{1(j,4,2)}$	$\tilde{r}_{(j,4,2)}$	$\tilde{i}_{(j,4,2)}$
	X_{43}		$\tilde{W}_{1(j,4,3)}$	$\tilde{r}_{(j,4,3)}$	$\tilde{i}_{(j,4,3)}$
	X_{44}		$\tilde{W}_{1(j,4,4)}$	$\tilde{r}_{(j,4,4)}$	$\tilde{i}_{(j,4,4)}$
X_5		$\tilde{W}_{2(j,5)}$			
	X_{51}		$\tilde{W}_{1(j,5,1)}$	$\tilde{r}_{(j,5,1)}$	$\tilde{i}_{(j,5,1)}$
	X_{52}		$\tilde{W}_{1(j,5,2)}$	$\tilde{r}_{(j,5,2)}$	$\tilde{i}_{(j,5,2)}$
X_6		$\tilde{W}_{2(j,6)}$			
	X_{61}		$\tilde{W}_{1(j,6,1)}$	$\tilde{r}_{(j,6,1)}$	$\tilde{i}_{(j,6,1)}$

iii. The second-stage assessment. Calculate the average defuzzified value of $W_{2A(h)}$ (denoted as $GW_{2A(h)}$).

iv. Calculate the final rate of aggregative risk RIK by the centroid defuzzified method as follows:

$$\begin{aligned}\text{RIK1} &= \sum_{j=1}^{n(DM)} GV(j) \times R_{2(j)} \\ &= \sum_j \sum_h \sum_k GV(j) \\ &\quad \times \left(\frac{GW_{2A(h)}}{\sum_{q=1}^{6} GW_{2A(q)}}\right)\left(\frac{GW_{1A(h,k)}}{\sum_{q=1}^{n(h)} GW_{1A(h,q)}}\right) \\ &\quad \times V(h,k,j)\end{aligned} \tag{12}$$

2. Lee's second algorithm (Algorithm II): This algorithm averages the rate individually and then averages the results to produce a final rate of aggregative risk. Algorithm II is very similar to the above algorithm. This algorithm is further detailed below:
   i. Calculate the rate of each project on the n decision makers' fuzzy data on each parameter.
   ii. The first-stage assessment. Establish a fuzzy assessment matrix for each attribute X_h, and then use these fuzzy assessment matrices to evaluate the first-stage aggregative assessment matrix $\tilde{R}_{1(h)}$ for each attribute X_h.
   iii. The second-stage assessment. Calculate the average defuzzified value of $W_{2A(h)}$(denoted as $GW_{2A(h)}$).
   iv. Evaluate the rate of aggregative risk for decision makers (D_j) first by

$$\text{RIK}(j) = \frac{\sum_{m=1}^{7} GV(m) R_{2(j,m)}}{\sum_{q=1}^{7} R_{2(j,q)}} = \sum_{m=1}^{7} GV(m) R_{2(j,m)} \tag{13}$$

Then average then, to obtain

$$\begin{aligned}\text{RIK2} &= \frac{1}{n}\sum_{j=1}^{n} RIK(j) \\ &= \frac{1}{n}\sum_m \sum_j \sum_h \sum_k GV(m) \\ &\quad \times \left(\frac{GW_{2(j,h)}}{\sum_{q=1}^{6} GW_{2(j,q)}}\right)\left(\frac{GW_{1(j,h,k)}}{\sum_{q=1}^{n(h)} GW_{1(j,h,q)}}\right) \\ &\quad \times V(j,h,k,m)\end{aligned} \tag{14}$$

4.1.2 Chen's [2] algorithms

Chen [2] proposed a new algorithm to evaluate the rate of aggregative risk in software development under a fuzzy group decision making environment. Chen also stated that this algorithm has the following advantages: (1) It does not need to form the fuzzy assessment matrices for attributes. (2) It does not need to perform complicated defuzzification operations of fuzzy numbers using the centroid method. This algorithm involves the following steps:

i. Chen first used a defuzzification method of trapezoidal fuzzy numbers to get the defuzzified value (denoted as e) of trapezoidal fuzzy numbers ($\tilde{M}$).
ii. Chen used the defuzzified method to convert fuzzy number representations of the weights, the grades of risk, and the grades of importance of risk items into real values.

Table 3 Linguistic values of grades of risk

Eleven ranks of grade of risk		
Notation	Linguistic Value	TFNs ($\tilde{r}_{(j,h,k)}$)
DL	Definitely low	(0.0, 0.0, 0.1)
EL	Extra low	(0.0, 0.1, 0.2)
VL	Very low	(0.1, 0.2, 0.3)
L	Low	(0.2, 0.3, 0.4)
SL	Slightly low	(0.3, 0.4, 0.5)
M	Middle	(0.4, 0.5, 0.6)
SH	Slightly high	(0.5, 0.6, 0.7)
H	High	(0.6, 0.7, 0.8)
VH	Very high	(0.7, 0.8, 0.9)
EH	Extra high	(0.8, 0.9, 1.0)
DH	Definitely high	(0.9, 1.0, 1.0)

Table 4 Linguistic values of relative importance

Five grades of relative importance		
Notation	Linguistic Value	TFNs ($\tilde{W}_{2(j,h)}/\tilde{W}_{1(j,h,k)}$)
1. VL	Very low	(0.0, 0.0, 0.25)
2. L	Low	(0.0, 0.25, 0.5)
3. M	Middle	(0.25, 0.5, 0.75)
4. H	High	(0.5, 0.75, 1.0)
5. VH	Very high	(0.75, 1.0, 1.0)

iii. Chen calculated average defizzified values of these real values for decision makers, and also calculated absolute weights of the risk items for each attribute.

iv. Finally, Chen calculated the final rate of aggregative risk RIK of the project by aggregating the risks of each attribute.

4.1.3 Adjusted algorithm for Example I

In [13], Lee uses 11 linguistic values for ranking the grades of risk items (see Table 3), which are represented by triangular fuzzy numbers. Furthermore, Lee also allows the experts to use five linguistic values $\{i.e.$ VL, L, M, H, VH$\}$ (see Table 4) represented by triangular fuzzy numbers (TFNs) for accessing the grades of importance of the risk items.

Decision makers can use either the importance set $W = \{$VL, L, M, H, VH$\}$ with the grade set $S = \{$VL, L, SL, M, SH, H, VH$\}$ or directly rating by normal triangular fuzzy numbers to access attribute weights, weights of risk items, and grades of risks.

To verify and compare the model with other methods, the result obtained here is compared with that obtained by Lee's algorithm [13] and Chen's algorithm [2] to validate the accuracy of the proposed model. Table 1 introduces the differences between their methods and the proposed model, and Sects. 4.1.1 and 4.1.2 describe these three algorithms.

The weights of MCDM, which are obtained by the proposed model, will be revised when a new expert joins or a decision maker chooses a different α value to fit the current situation. To verify the proposed model, this study assumes that a symbol D($\tilde{A}$) denotes the defuzzification result of this fuzzy number $\tilde{A}$ by the centroid method [13], and uses the symbol (Table 2) in the research of Lee [13] as an example to explain each step. The adjusted steps based on section 3.2's algorithm are as follows:

Step 1 Build hierarchical structure model from determination problem and number of attributes (N).
For example, Lee [13] *presented the hierarchical structure model of aggregative risk along with attributes $N = 6$.*

Step 2 Obtain opinions of experts in software development, and then collect their evaluative attribute weights in respect to the hierarchical structure model (see Table 5).

Step 3 List the feasible projects, and request the experts to evaluate the grades of these projects in respect to the risk items (see Table 6).

Table 5 The weights of three projects for two decision makers [13]

Attribute	Risk item	Weight-2		Weight-1	
		D_1	D_2	D_1	D_2
X_1		(0.1, 0.25, 0.35)	(0.15, 0.3, 0.5)		
	X_{11}			(0.7, 0.85, 1)	(0.8, 0.9, 1)
X_2		(0.3, 0.5, 0.6)	(0.25, 0.4, 0.6)		
	X_{21}			(0.2, 0.3, 0.4)	(0.2, 0.35, 0.45)
	X_{22}			(0.3, 0.4, 0.5)	(0.1, 0.3, 0.5)
	X_{23}			(0.15, 0.3, 0.4)	(0.2, 0.3, 0.4)
	X_{24}			(0.2, 0.35, 0.5)	(0.1, 0.3, 0.5)
X_3		(0.2, 0.3, 0.4)	(0.1, 0.2, 0.3)		
	X_{31}			(0.1, 0.15, 0.25)	(0.35, 0.55, 0.85)
	X_{32}			(0.35, 0.5, 0.6)	(0.25, 0.55, 0.65)
X_4		(0.2, 0.35, 0.5)	(0.2, 0.3, 0.4)		
	X_{41}			(0.3, 0.4, 0.5)	(0.2, 0.3, 0.4)
	X_{42}			(0.1, 0.2, 0.3)	(0.1, 0.2, 0.3)
	X_{43}			(0.2, 0.3, 0.4)	(0.2, 0.3, 0.4)
	X_{44}			(0.2, 0.3, 0.4)	(0.2, 0.3, 0.4)
X_5		(0.1, 0.25, 0.4)	(0.1, 0.3, 0.4)		
	X_{51}			(0.5, 0.6, 0.7)	(0.4, 0.5, 0.6)
	X_{52}			(0.4, 0.5, 0.6)	(0.6, 0.7, 0.8)
X_6		(0.1, 0.3, 0.5)	(0.15, 0.3, 0.5)		
	X_{61}			(0.8, 0.9, 1)	(0.9, 1, 1)

Table 6 The grades of risk of three projects for two decision makers [13]

Attribute	Risk item	Grade of risk	
		D$_1$	D$_2$
X_1			
	X_{11}	(I) (0.4, 0.5, 0.6)	(0.5, 0.6, 0.7)
		(II) (0.6, 0.7, 0.8)	(0.7, 0.8, 0.9)
		(III) (0, 0.1, 0.2)	(0, 0, 0.1)
X_2			
	X_{21}	(I) (0.2, 0.3, 0.4)	(0.2, 0.3, 0.4)
		(II) (0.6, 0.7, 0.8)	(0.8, 0.9, 1)
		(III) (0.1, 0.2, 0.3)	(0.2, 0.3, 0.4)
	X_{22}	(I) (0.3, 0.4, 0.5)	(0.2, 0.4, 0.6)
		(II) (0.5, 0.6, 0.7)	(0.5, 0.6, 0.7)
		(III) (0, 0, 0.1)	(0, 0.1, 0.2)
	X_{23}	(I) (0.2, 0.4, 0.5)	(0.3, 0.5, 0.6)
		(II) (0.7, 0.8, 0.9)	(0.7, 0.8, 0.9)
		(III) (0.1, 0.2, 0.3)	(0, 0.1, 0.2)
	X_{24}	(I) (0.5, 0.6, 0.7)	(0.1, 0.2, 0.3)
		(II) (0.6, 0.7, 0.8)	(0.8, 0.9, 1)
		(III) (0.1, 0.2, 0.3)	(0.2, 0.3, 0.4)
X_3			
	X_{31}	(I) (0.25, 0.35, 0.45)	(0.35, 0.45, 0.55)
		(II) (0.7, 0.8, 0.9)	(0.7, 0.8, 0.9)
		(III) (0.1, 0.2, 0.3)	(0.1, 0.2, 0.3)
	X_{32}	(I) (0.4, 0.6, 0.8)	(0.2, 0.4, 0.6)
		(II) (0.5, 0.6, 0.7)	(0.5, 0.6, 0.7)
		(III) (0.2, 0.3, 0.4)	(0.1, 0.2, 0.3)
X_4			
	X_{41}	(I) (0.2, 0.3, 0.4)	(0.25, 0.4, 0.55)
		(II) (0.6, 0.7, 0.8)	(0.7, 0.8, 0.9)
		(III) (0.2, 0.3, 0.4)	(0.2, 0.3, 0.4)
	X_{42}	(I) (0.1, 0.2, 0.3)	(0.2, 0.3, 0.4)
		(II) (0.6, 0.7, 0.8)	(0.7, 0.8, 0.9)
		(III) (0.1, 0.2, 0.3)	(0, 0.1, 0.2)
	X_{43}	(I) (0.3, 0.4, 0.5)	(0.3, 0.4, 0.5)
		(II) (0.7, 0.8, 0.9)	(0.7, 0.8, 0.9)
		(III) (0.1, 0.2, 0.3)	(0.2, 0.3, 0.4)
	X_{44}	(I) (0.2, 0.3, 0.4)	(0.2, 0.3, 0.4)
		(II) (0.6, 0.7, 0.8)	(0.6, 0.7, 0.8)
		(III) (0.2, 0.3, 0.4)	(0.1, 0.2, 0.3)
X_5			
	X_{51}	(I) (0.2, 0.3, 0.4)	(0.2, 0.3, 0.4)
		(II) (0.5, 0.6, 0.7)	(0.6, 0.7, 0.8)
		(III) (0.2, 0.3, 0.4)	(0.1, 0.2, 0.3)
	X_{52}	(I) (0.3, 0.4, 0.5)	(0.3, 0.4, 0.5)
		(II) (0.7, 0.8, 0.9)	(0.7, 0.8, 0.9)
		(III) (0.2, 0.3, 0.4)	(0.1, 0.2, 0.3)
X_6			
	X_{61}	(I) (0.4, 0.5, 0.6)	(0.4, 0.5, 0.6)
		(II) (0.5, 0.6, 0.7)	(0.5, 0.6, 0.7)
		(III) (0.1, 0.2, 0.3)	(0.2, 0.3, 0.4)

Step 4 If no new expert is available, execute step 5. Otherwise, if the experts do not have significant orderings, assign the same weight for evaluation. Meanwhile, perform the OWA aggregation process to obtain the weights for evaluation, and then execute steps 2 and 3 by the result of this step. (*This study assumes that these two experts have equal weights, denoted as* $W_{e1} = W_{e2} = 0.5$).

Step 5 The weights of each expert are used to multiply their evaluative attribute weights. The Centroid method [13] then is used to defuzzify the aggregation result of the attribute weights.

This step can be divided into two branch steps——

Step 5.1 Aggregate the attribute weights according to each expert (The aggregation weight of attribute h (denoted as $\tilde{W}_{2A(h)}$) is calculated by using equation (15)).

$$\text{Let } \tilde{W}_{2A(h)} = \frac{1}{n}(\times)\left[\tilde{W}_{2(1,h)} (+)\tilde{W}_{2(2,h)}(+)\cdots\cdots,(+)\tilde{W}_{2(n,h)}\right] = (a_{2A(h)}, b_{2A(h)}, c_{2A(h)}) \tag{15}$$

$$\text{where } a_{2A(h)} = \frac{1}{n}\sum_{j=1}^{n} a_{2(j,h)},\ b_{2A(h)} = \frac{1}{n}\sum_{j=1}^{n} b_{2(j,h)},\ \text{and } c_{2A(h)} = \frac{1}{n}\sum_{j=1}^{n} c_{2(j,h)}$$

Step 5.2 The Centroid method [13] is used to defuzzify these weights.

The Centroid method is used to defuzzify any kind of fuzzy sets, such as triangular fuzzy numbers, trapezoidal fuzzy numbers, and so on. In the present example, only triangular fuzzy number is used. The defuzzified value of the weight of attribute h thus is

$$D\left(\tilde{W}_{2A(h)}\right) = \frac{1}{3} \times (a_{2A(h)} + b_{2A(h)} + c_{2A(h)}) \tag{16}$$

Step 6 Sort the defuzzified attribute weights and execute OWA aggregation to obtain refined attribute weights (denoted as $W'_{2A(1)} \sim W'_{2A(m)}$). The results are shown in Table 7.

This step can be divided into two branch steps——

Step 6.1 Choose an appropriate sorting method to sort the defuzzified attribute weights.

Step 6.2 Use equations (5)–(7) to obtain the OWA weights, and replace the attribute weights with these refined weights according to the sorting result. The distribution of the refined weights in Example I under different α's values is shown in Fig. 2.

Step 7 Like the process of step 5, the aggregative weights of each risk items are obtained by equation (17) (denoted as $\tilde{W}_{1A(h,k)}$), and these aggregative weights are defuzzified in equation (16) (denoted as $D(\tilde{W}_{1A(1,1)}) \sim D(\tilde{W}_{1A(6,1)})$). Equation (19) then is used to distribute the refined weight(s) of the risk item(s) of each attribute based on the ratio of weights of these risk item(s) given by the experts (denoted as $W'_{1A(1,1)}$

Table 7 The weights of attributes after OWA aggregation

	$\alpha = 0.5$	$\alpha = 0.6$	$\alpha = 0.7$	$\alpha = 0.8$	$\alpha = 0.9$	$\alpha = 1.0$
Personnel (W_1)	0.16666	0.14614	0.11416	0.07229	0.02548	0
System requirement (W2)	0.16666	0.24676	0.34747	0.47811	0.66372	1
Schedules and budgets (W_3)	0.16666	0.10308	0.05437	0.02053	0.00290	0
Developing technology (W_4)	0.16666	0.20721	0.23977	0.25473	0.22396	0
External resource (W_5)	0.16666	0.12274	0.07876	0.03851	0.00860	0
Performance (W_6)	0.16666	0.17401	0.16543	0.13571	0.07559	0

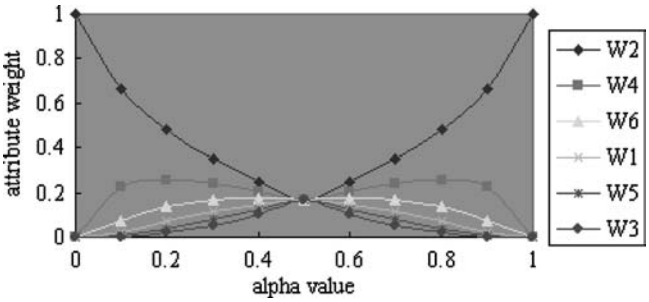

Fig. 2 The distribution of the refined weights in Example I

$\sim W'_{1A(6,1)}$). The results are shown in Table 8.

$$\text{Let } \tilde{W}_{1A(h,k)} = \frac{1}{n}(\times)\left[\tilde{W}_{1(1,h,k)}(+)\tilde{W}_{1(2,h,k)}(+)\cdots\cdots,(+)\tilde{W}_{1(n,h,k)}\right] = (a_{1A(j,h,k)}, b_{1A(j,h,k)}, c_{1A(j,h,k)}) \quad (17)$$

$$w'_{ij} = \frac{w_{ij}}{\sum_j w_{ij}} \times w_i, \quad i = \text{ number of attributes}; \quad j = \text{ number of risk items} \quad (18)$$

$$\text{So, } W'_{1A(h,k)} = \frac{D(\tilde{W}_{1A(h,k)})}{\sum_{j=1}^{n(h)} D(\tilde{W}_{1A(h,j)})} \times W'_{2A(h)} \quad (19)$$

$$\text{where } \sum_{j=1}^{n(h)} D(W_{1A(h,j)}) \neq 0$$

Step 8 Multiply the weights of the risk items by their project grades, and then rank their orderings to make reference solution to the decision maker.

This step can be divided into two branch steps——

Step 8.1 Calculate the aggregative risk value of each risk item towards a select project using equation (20).

$$\tilde{r}_{A(h,k)} = \frac{1}{n}(\times)[\tilde{r}_{(1,h,k)}(+)\tilde{r}_{(2,h,k)}(+)\cdots\cdots,(+)\tilde{r}_{(n,h,k)}] = (a_{3A(h,k)}, b_{3A(h,k)}, c_{3A(h,k)}), \quad (20)$$

where $a_{3A(h,k)} = \frac{1}{n}\sum_{j=1}^{n} a_{3(j,h,k)}$,

$b_{3A(h,k)} = \frac{1}{n}\sum_{j=1}^{n} b_{3(j,h,k)}$, $c_{3A(h,k)} = \frac{1}{n}\sum_{j=1}^{n} c_{3(j,h,k)}$

Their defuzzified values then are:

$$D\left(\tilde{r}_{A(h,k)}\right) = \frac{1}{3}\left[a_{3A(h,k)} + b_{3A(h,k)} + c_{3A(h,k)}\right] \quad (21)$$

Step 8.2 Multiply the refined weights of risk items by their project grades. Finally, the aggregative result of project P is

$$Agg\_Result(P) = \sum_{h=1}^{n}\sum_{k=1}^{n(h)}\left[W'_{1A(h,k)} \times D(r_{A(h,k)})\right] \quad (22)$$

where $P = 1, 2, \ldots, n(P)$, n is the number of attributes, $n(h)$ is the number of risk items, and $n(P)$ is the number of projects.

4.1.4 Results for Example I

Because the proposed system has the same results of $\alpha = 0.5 + \delta$ and $\alpha = 0.5 - \delta$ $(0 \leq \delta \leq 0.5)$, it merely shows the data of $\alpha \geq 0.5$ to represent the total result. For example, the aggregation results are the same when $\alpha = 0.7$ and $\alpha = 0.3$. Besides, even if parameter α is a continuous value in interval [0,1], this study merely illustrates the output values when $\alpha = 0.5$, 0.6, 0.7, 0.8, 0.9, and 1.0. According to the entropy of information after OWA aggregation of the input data, the output weights of the attributes and risk items are summarized in Tables 7 and 8, respectively. From the last column in Table 7, the proposed model can be viewed as a magnifying lens to determine the most important attribute based on the situation of sparest information (*i.e.* optimistic and $\alpha = 0$ or 1). In the second column of Table 7, when $\alpha = 0.5$ (moderate situation), the proposed model can obtain the attribute weights (equal weights) based on maximum information.

Similarly, we take α value from 0.5 to 1.0 as the parameter for execution according to our algorithm for the purpose of verification in respect to proposed model, and the results are presented below in Table 9.

To verify the validity of the proposed model, this study compares the result of the proposed algorithm with the algorithms in Lee [13] and Chen [2]. However, the aggregation results will be changed corresponding with α's value, thus the extreme values of α are chosen for representation purposes. Therefore, Table 10 selects two extreme points to summarize the aggregation results of three projects (i.e. maximum (α=0.5) and minimum information (α=1 or 0) entropy).

Table 10 reveals that the rank of the proposed algorithm is the risk of Project (II) > Project (I) > Project (III), which

Table 8 The weights of risk items after OWA aggregation

Attribute	Risk item	$\alpha = 0.5$	$\alpha = 0.6$	$\alpha = 0.7$	$\alpha = 0.8$	$\alpha = 0.9$	$\alpha = 1.0$
Personnel							
	Personnel shortfalls, key person(s) quit (W_{11})	0.16666	0.14614	0.11416	0.07229	0.02548	0
System requirement							
	Requirement ambiguity (W_{21})	0.04112	0.06098	0.08574	0.11797	0.16377	0.24675
	Developing the wrong software function (W_{22})	0.04545	0.06729	0.09476	0.13039	0.18101	0.27272
	Developing the wrong user interface (W_{23})	0.03787	0.05608	0.07897	0.10866	0.15084	0.22727
	Continuing stream requirement changes (W_{24})	0.04220	0.06249	0.08799	0.12108	0.16808	0.25324
Schedules and budgets							
	Schedule not accurate (W_{31})	0.07281	0.04503	0.02375	0.00897	0.00126	0
	Budget not sufficient (W_{32})	0.09385	0.05804	0.03061	0.01156	0.00163	0
Developing technology							
	Gold-plating (W_{41})	0.05072	0.06306	0.07297	0.07752	0.06816	0
	Skill levels inadequate (W_{42})	0.02898	0.03603	0.04170	0.04430	0.03895	0
	Straining hardware (W_{43})	0.04347	0.05405	0.06255	0.06645	0.05842	0
	Straining software (W_{44})	0.04347	0.05405	0.06255	0.06645	0.05842	0
External resource							
	Shortfalls in externally furished components (W_{51})	0.07971	0.05870	0.03767	0.01841	0.00411	0
	Shortfalls in externally performed tasks (W_{52})	0.08695	0.06403	0.04109	0.02009	0.00449	0
Performance							
	Real-time performance shortfalls (W_{61})	0.16666	0.17401	0.16543	0.13571	0.07559	0

Table 9 The aggregation result of our algorithm

	$\alpha = 0.5$	$\alpha = 0.6$	$\alpha = 0.7$	$\alpha = 0.8$	$\alpha = 0.9$	$\alpha = 1.0$
Project (I)	0.42598	0.41974	0.41098	0.39971	0.38725	0.38668
Project (II)	0.71018	0.71362	0.71848	0.72527	0.73548	0.74545
Project (III)	0.20332	0.20254	0.20208	0.20094	0.19657	0.17727

Table 10 The evaluation results for Example I

Algorithm	Project		
	(I)	(II)	(III)
Algorithm 1 by Lee [13]	0.19650	0.50917	0.09112
Algorithm 2 by Lee [13]	0.19648	0.50930	0.09173
Algorithm by Chen [2]	0.18541	0.51086	0.05767
Our proposed algorithm ($\alpha = 0.5$)	0.42598	0.71018	0.20332
Our proposed algorithm ($\alpha = 1$ or 0)	0.38668	0.74545	0.17727

is the same as the ratings of the projects calculated by the algorithm of Lee [13] and Chen [2]. The proposed model thus can be validated.

4.2 Example II: evaluation of distribution centers in logistic

In this section, we adopted an example introduced by Chen [3]. This example discusses the distribution center selection through the evaluation of external performance in one convenience store of Taiwan.

4.2.1 Chen's [3] algorithm

The membership function for linguistics values is the same as definition in Table 4. Chen [3] proposed an algorithm of multi-person multi-criteria external performance evaluation in logistics with fuzzy approach can be expressed by the following steps:

Step 1 Construction of hierarchical structure
Step 2 Evaluate the importance weight of extracted criterion (use fuzzy Delphi method)
Step 3 Construction of linguistic scales for linguistic variables
Step 4 Aggregation of fuzzy appropriateness indices
Step 5 Computation of fuzzy overall evaluation
Step 6 Defuzzificaion of fuzzy overall evaluation
Step 7 Analysis and decision

4.2.2 Adjusted algorithm for Example II

To verify the proposed model, we use the data in Chen [3] as an example to explain each step, and the adjusted steps based on section 3.2's algorithm are as follows:

Step 1 Build hierarchical structure model from determination problem and number of attributes (N).
After factor analysis, six criteria were extracted: (1) Efficiency (C_1); (2) Customer (C_2); (3) Stockouts (C_3); (4) Delivery (C_4); (5) Order (C_5); (6) Personnel (C_6). [3]

Step 2 Obtain opinions of domain experts and then collect their evaluative attribute weights of attributes in respect to the hierarchical structure model (see Table 11 [3]).

Step 3 List the feasible alternatives, and request the experts to evaluate the grades of these projects.
The six commonly used distribution centers in this case are: A_1 = Wong Chuan Logistics Corp., A_2 = Da Je Tong Lo-Support International, A_3 = Shen Hong
Logistics Corp., A_4 = Retail Support International, A_5 = Ta Jung Transportation and A_6 = Chiao Tai

Table 11 The aggregative importance of criteria

Criteria	Weight (Triangular Fuzzy Number)
Efficiency (C_1)	$\tilde{W}_1 = (0.2887, 0.2992, 0.3118)$
Customer (C_2)	$\tilde{W}_2 = (0.1047, 0.1204, 0.1348)$
Stockouts (C_3)	$\tilde{W}_3 = (0.1947, 0.2040, 0.2099)$
Delivery (C_4)	$\tilde{W}_4 = (0.0425, 0.0590, 0.1086)$
Order (C_5)	$\tilde{W}_5 = (0.0425, 0.0480, 0.0507)$
Personnel (C_6)	$\tilde{W}_6 = (0.2325, 0.2639, 0.2975)$

Table 12 The fuzzy appropriateness indices of the six alternatives under each criterion [3]

R_i	Alternatives	
	A_1	A_2
C_1	(0.4168, 0.8025, 0.9714)	(0.4011, 0.7814, 0.9571)
C_2	(0.4156, 0.8000, 0.9506)	(0.3950, 0.7731, 0.9256)
C_3	(0.3756, 0.7444, 0.9500)	(0.3688, 0.7344, 0.9506)
C_4	(0.3600, 0.7225, 0.9750)	(0.3913, 0.7663, 0.9750)
C_5	(0.4200, 0.8075, 1.0000)	(0.4550, 0.8550, 1.0000)
C_6	(0.2875, 0.6188, 0.9250)	(0.3538, 0.7163, 0.9500)
	A_3	A_4
C_1	(0.3893, 0.7643, 0.9650)	(0.4211, 0.8086, 0.9786)
C_2	(0.4044, 0.7856, 0.9500)	(0.3994, 0.7775, 0.9381)
C_3	(0.3606, 0.7231, 0.9256)	(0.3913, 0.7663, 0.9506)
C_4	(0.3913, 0.7663, 0.9750)	(0.3763, 0.7450, 0.9500)
C_5	(0.4200, 0.8075, 1.0000)	(0.4200, 0.8075, 1.0000)
C_6	(0.3413, 0.6975, 0.9500)	(0.3263, 0.6763, 0.9500)
	A_5	A_6
C_1	(0.3939, 0.7707, 0.9650)	(0.3896, 0.7646, 0.9507)
C_2	(0.3800, 0.7519, 0.9381)	(0.3931, 0.7713, 0.9375)
C_3	(0.3675, 0.7331, 0.9506)	(0.3375, 0.6906, 0.9506)
C_4	(0.4075, 0.7888, 0.9750)	(0.2975, 0.6350, 0.9250)
C_5	(0.3850, 0.7600, 0.9500)	(0.4200, 0.8075, 1.0000)
C_6	(0.3263, 0.6763, 0.9250)	(0.3138, 0.6575, 0.9250)

Logistics Corp. The grades of these alternatives under each criterion are shown in Table 12 [3].

Step 4 If no new expert is available, execute step 5. If the experts do not have significant orderings, assign equal weight for evaluation. Otherwise, perform the OWA aggregation process to obtain the weights of experts for evaluation.

Step 5 The weights of each expert multiply their evaluative attribute weights to form the aggregative weights of attributes. (*In this example, steps 4 and 5 can be skipped.*)

Step 6 Sort the attribute/criteria weights and execute OWA aggregation (by equation (5)–(7)) to obtain refined attribute weights.
The ranking order of the defuzzified values in Table 11 is $\tilde{W}_1 > \tilde{W}_6 > \tilde{W}_3 > \tilde{W}_2 > \tilde{W}_4 > \tilde{W}_5$. Then, the distribution of refined weights for each criterion after OWA is shown in Fig. 3.

Step 7 If the aggregative weights of sub-attributes are exist, to distribute the refined weight(s) of the sub-attributes of each attribute based on the ratio of weights of these sub-attributes given by the experts. *(In this example, this step can be skipped.)*

Step 8 Multiply the weights of the attributes by their project grades, and then rank their orderings to make reference solution to the decision maker.
The ranking orders and defuzzified values of the fuzzy overall evaluation for each alternative based on the algorithms of proposed model ($\alpha = 0.5, 0.6, 0.7, 0.8, 0.9, 1.0$) and Chen's method are shown in Table 13. The overall evaluation values of alternatives under different α's values is as Fig. 4. The ranking orders of proposed method will be consistent with Chen's method under $\alpha = [0.63, 0.78]$.

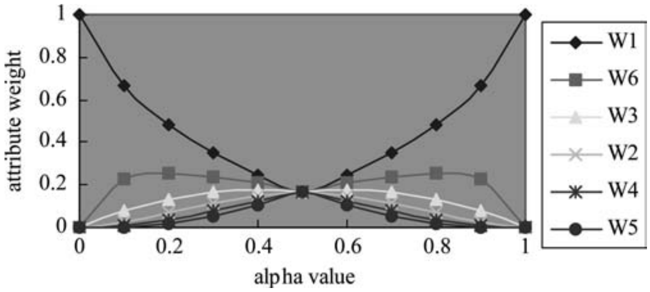

Fig. 3 The distribution of the refined weights in Example II

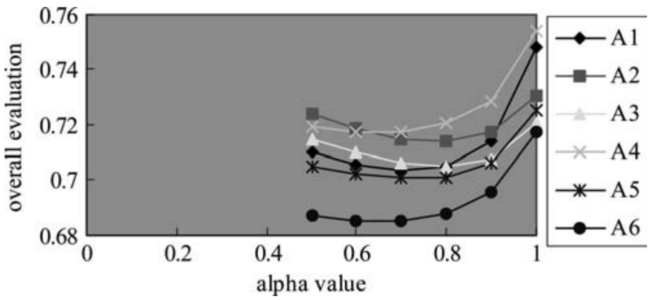

Fig. 4 The overall results under $\alpha = [0.5, 1.0]$

5 Discussion

After the experiments in Sect. 4, we find the proposed model having the following characteristics:

A. Sensitivity and Robustness

Due to the evaluating grades of risk for each project are clearly having trend (i.e. Project (II) > Project (I) > Project (III) in almost risk items), we find that the ranking order of projects in Example I is robust under different α's values (see Table 9). However, the ranking order in Example II will change based on different α's values (see Table 13 and Fig. 4). This is because the evaluating values of criteria for several alternatives are approximate in the in Example II (see Table 12), and the overall evaluating results will easily affect by the refined weights (OWA weights).

B. Effect of "magnifying lens"

From Table 7 and Figs. 2 and 3, if the α value changes from 0.5 to 1.0 (or 0), the weights of attributes will be changed from

Table 13 The evaluation results for Example II

		Overall evaluation values						Ranking order
		A_1	A_2	A_3	A_4	A_5	A_6	
Proposed method	$\alpha = 0.5$	0.7099	0.7240	0.7150	0.7191	0.7052	0.6872	$A_2 > A_4 > A_3 > A_1 > A_5 > A_6$
	$\alpha = 0.6$	0.7059	0.7185	0.7099	0.7175	0.7024	0.6851	$A_2 > A_4 > A_3 > A_1 > A_5 > A_6$
	$\alpha = 0.7$	0.7039	0.7150	0.7063	0.7177	0.7008	0.6851	$A_4 > A_2 > A_3 > A_1 > A_5 > A_6$
	$\alpha = 0.8$	0.7052	0.7140	0.7047	0.7204	0.7011	0.6877	$A_4 > A_2 > A_1 > A_3 > A_5 > A_6$
	$\alpha = 0.9$	0.7141	0.7174	0.7075	0.7288	0.7061	0.6954	$A_4 > A_2 > A_1 > A_3 > A_5 > A_6$
	$\alpha = 1.0$	0.7483	0.7303	0.7207	0.7542	0.7251	0.7174	$A_4 > A_1 > A_2 > A_5 > A_3 > A_6$
Chen's method [3]		0.6109	0.6236	0.6147	0.6256	0.6089	0.5940	$A_4 > A_2 > A_3 > A_1 > A_5 > A_6$

distributed (equal weights) to centralized (the most important attribute).

When dealing with the problems in management, we usually just need to face the key problem, which can help us with overcoming the difficult situation. So, we can use this model to search out the critical attribute of the problem when the α value given by the project manager is 0 or 1 (*i.e.* under the minimal entropy). Therefore, the proposed model would be a useful tool if the project manager wants to find the most important attribute (or criterion).

After analysis the results of Examples I and II, we suggest the decision maker can adjust α's value under the following situation:

1. No preference: When a decision maker has no preference toward the criteria, we can assign these attributes equal weight. Under this circumstance, the suggesting α's value is 0.5.
2. Partial preference: We suggest the range of α's values is [0.6, 0.9], when a decision maker has collected criteria weights from domain experts and want to execute sensitivity analysis for making final decision based the opinions of experts.
3. Single preference: If the decision maker is confident and believe in the most important criterion, we suggest to assign $\alpha = 1.0$. This can enlarge the effect of this single preference criterion.

6 Conclusion

This study has proposed a dynamic fuzzy OWA model to deal with problems of group multiple criteria decision making. The proposed model can help users to solve MCDM problems under the situation of fuzzy or incomplete information. The advantages of this study are:

(1) The proposed approach can modify associated dynamical weights based on the aggregation situation (information capacity).
(2) The fuzzy linguistic variables are used to help the decision maker to obtain the criteria weights more flexibly and reasonably (based on situation).
(3) The fuzzy OWA model can work like a "magnifying lens" to enlarge/find the most important attribute, which is dependent on the sparest information (*i.e.* optimistic case: situation parameter $\alpha = 0$ or 1), or obtain equal weights of attributes based on maximal information (*i.e.*moderate case: situation parameter $\alpha = 0.5$).

Future studies may find that the applications of some other group MCDM problems. Or, the proposed model can be adapted to fuzzy neuron network (FNN) for achieving a better solution to uncertain problems. Fuzzy rule base of FNN can be combined with the fuzzy OWA model presented here, which can help accelerate the convergence of the fuzzy rule base.

References

1. Borcherding K, Epple T, Winterfeldt DV (1991) Comparison of weighting judgments in multi-attribute utility measurement. Manage Sci 37(12):1603–1619
2. Chen SM (2001) Fuzzy group decision making for evaluating the rate of aggregative risk in software development. Fuzzy Sets Syst 118:75–88
3. Chen YC (2002) An application of fuzzy set theory to the external performance evaluation of distribution centers in logistics. Soft Comput 6:64–70
4. Choi DY (1999) A new aggregation method in a fuzzy environment. Decis Support Syst 25:39–51
5. Choo EU, Wedley WC (1985) Optimal criteria weights in repetitive multicriteria decision making. J Oper Res Soc 37:527–541
6. Darmon RY, Rouzies D (1991) Internal validity assessment of conjoint estimated attribute importance weights. J Acad Marketing Sci 19(4):315–322
7. Dyer JS, Sarin RK (1979) Measurable multiattribute functions. Oper Res 27:810–822
8. Fuller R, Majlender P (2001) An analytic approach for obtaining maximal entropy OWA operator weights. Fuzzy Sets Syst 124:53–57
9. Jaynes ET (1989) Cleaning up mysteries: the original goal, maximum entropy and Bayesian methods. Kluwer, Dordrecht
10. Jaynes ET, Rosen krantz RD (1983) Papers on probability, statistics and statistical physics. D. Reidel, Dordrecht
11. Klir GJ (1988) Fuzzy sets, uncertainly and information. Prentice Hall, Englewood Cliffs
12. Klir GJ, Wierman MJ (1999) Uncertainty-based information, 2nd edn. Physica-Verlag, Germany
13. Lee HM (1996) Group decision making using fuzzy sets theory for evaluating the rate of aggregative risk in software development. Fuzzy Sets Syst 80:261–271
14. Mendel JM (2000) Uncertain rule-based fuzzy logic systems: introduction and new directions. Prentice Hall PTR, Upper Saddle River
15. Mesiar R, Saminger S (2004) Domination of ordered weighted averaging operators over t-norms. Soft Comput 8:562–570
16. Moshkovich HM, Schellenberger RE, Olson DL (1998) Data influences the result more than preferences: some lessons from implementation of multiattribute techniques in a real decision task. Decis Support Syst 22:73–84

17. Nutt PC (1980) Comparing methods for weighting decision criteria. OMEGA 8:163–172
18. O'Hagan M (1988) Aggregating template or rule antecedents in real-time expert systems with fuzzy set logic. In: Proceedings of 22nd Annual IEEE Asilomar conference on signals, systems, computers, Pacific Grove, pp 681–689
19. Pekelman D, Sen S (1974) Mathematical programming models for determination of attribute weights. Manage Sci 20(8):1217–1229
20. Ribeiro RA, Pereira RAM (2003) Generalized mixture operators using weighting functions: a comparative study with WA and OWA. Eur J Oper Res 145:329–342
21. Shannon CE (1948) A mathematical theory of communication. Bell Syst Tech J 27:379–423
22. Shoemaker PJH, Carter WC (1982) An experimental comparison of different approaches to determining weights in additive utility models. Manage Sci 28:182–196
23. Smolikova R, Wachowiak MP (2002) Aggregation operators for selection problems. Fuzzy Sets Syst 131:23–34
24. Solymosi T, Dombi J (1986) A method for determining the weights of criteria: the centralized weights. Eur J Oper Res 26:35–41
25. Weber M, Eisenfhr F, von Winterfeldt D (1988) The effects of splitting attributes on weights in multiattribute utility measurement. Manage Sci 34:431–445
26. Yager RR (1980) On a general class of fuzzy connectives. Fuzzy Sets Syst 4:235–242
27. Yager RR (1988) Ordered weighted averaging aggregation operators in multi-criteria decision making. IEEE Trans Syst Man Cybern 18:183–190
28. Yager RR (1991) Connectives and quantifiers in fuzzy sets. Fuzzy Sets Syst 40:39–75
29. Yager RR, Kacprzyk J (1997) The ordered weighted averaging operators. Kluwer, Boston
30. Zadeh LA (1965) Fuzzy sets. Inf Control 8:338–353
31. Zhang D, Yu PL, Wang PZ (1992) State-dependent weights in multicriteria value functions. J Optimization Theory Appl 74(1):1–21
32. Zimmermann HJ, Zysno P (1980) Latent connectives in human decision making. Fuzzy Sets Syst (North-Holland) 4:37–51