DYNAMIC FUZZY OWA GDSS FOR EVALUATING THE RISKS OF SOFTWARE DEVELOPMENT

Ching-Hsue Cheng\textsuperscript{a} Jing-Rong Chang\textsuperscript{a} Ling-Show Chen\textsuperscript{b}

\textsuperscript{a}Department of Management Information Systems, National Yunlin University of Science and Technology, Touliu 640, Yunlin, Taiwan
\textsuperscript{b}Department of Information Management, Kun Shan University of Technology 949, Da Wan Road, Yung Kang, Tainan 710, Taiwan
E-Mail: chcheng@pine.yuntech.edu.tw (Ching-Hsue Cheng)

ABSTRACT

It is difficult to deal with the traditional weight problems of MCDM (multiple criteria decision making). However, the previous aggregation operators (include OWA) do not take into account the situational factors when processes the aggregation, \textit{i.e.} these operators and aggregation situation are independent.

In order to solve this problem mentioned above, we propose a new powerful OWA aggregation model which can modify associated weights dynamical based on the aggregation situation and work like a “magnifying glass” to find the most important attribute dependent on the sparsest information \((\text{i.e.} \alpha=0 \text{ or } 1)\), or get equal attributes’ weights based on maximal information \((\text{i.e.} \alpha=0.5)\). For illustrating and verifying this new model we developed a Group Decision Support System (GDSS) to evaluate the risks of software and compared the running results with Chen’s \cite{13} and Lee’s \cite{6} method.

Keywords: Fuzzy Group Decision Support System (Fuzzy GDSS), Multiple Criteria Decision Making (MCDM), Ordered Weighted Averaging (OWA).

1. INTRODUCTION

When facing the decision problems, we usually use some significant criteria to evaluate these alternatives and make the final decision. In traditional methods, we always dealt it with crisp value. In practical, it is not easy to present criteria or attributes by crisp value, so the concept of fuzzy theory is proper to add to solve the MCDM problems.

Yager \cite{12} first introduced the concept of OWA operators to solve the MCDM problem. Numbers of approaches have been proposed to calculate the weights and O’Hagan \cite{9} is the first one to use the concept of entropy in the OWA operation, but those methods still did not take into account the factor of situation. In order to resolve this problem we propose a powerful OWA aggregation model based on the faster OWA operator, which was introduced by Fuller and Majlender \cite{10} and it can work like a magnifying glass to adjust its focus based on the sparsest information to change the weights of attributes dynamically, and revise the weight of each attributes based on aggregation situation and then provide suggestion to decision maker.

We divide this paper into six sections. Section 2 presents a basic concept of Yager’s \cite{12} OWA operator and section 3 introduces how we use Yager’s OWA operator to fast aggregate attributes’ weights based on the information entropy. In section 4, we introduce the significant concepts and algorithm of our proposed model. Section 5 develops an OWA GDSS and then we illustrate a numerical example to verify and compare this model with other methods by an OWA GDSS in section 6. Finally, we give some advantages of this new model and conclusion at the end of this paper.

2. THE OWA OPERATOR

Yager \cite{12} propose an order weighted averaging (OWA) operator which has the ability that it can get the optimal weights of the attributes based on the rank of these weighting vectors after the aggregation process (reference to Definition 1.).

Definition 1. An OWA operator of dimension \(n\) is a mapping \(F: \mathbb{R}^n \rightarrow \mathbb{R}\), that has an associated weighting vector \(W\), of having the properties

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

and such that

\[
f(\alpha_1, \ldots, \alpha_n) = \sum_{j=1}^n w_j b_j
\] (1)
where $b_j$ is the $j$th largest element of the collection of the aggregated objects \{ $a_1,...,a_n$ \}

Yager [12] also introduces two important characterizing measures. The first one is the measure of orness of the aggregation, which defined as

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

And, the second one, imply the measure of dispersion of the aggregation, is defined as

$$\text{Disp}(W) = \sum_{i=1}^{n} W_i \ln W_i$$ (3)

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

3. USE MAXIMAL ENTROPY TO OBTAIN THE OWA WEIGHTS

O’Hagan [9] suggests a method which combines the maximum entropy principle [5] and Yager’s approach [12] to determine a special class of OWA operators having the maximal entropy [2] of the OWA weights for a given level of orness. This approach is based on the solution of the following problem which are similar to formula (1)~(3):

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

Subject to the constraints

$$\alpha = \frac{1}{n-1} \sum_{i=1}^{n} (n-i) W_i, \quad 0 \leq \alpha \leq 1$$ (4)

Fuller and Majlender [10] use the method of Lagrange multipliers to transfer equation (4) which can determine the optimal weighting vector, and the associated weighting vector is easily obtained by (5)~(7).

$$\ln W_i = \frac{j-1}{n-1} \ln W_{i-1} + \frac{n-j}{n-1} \ln W_n \Rightarrow W_i = \sqrt[n-1]{W_n W_{i-1}}^{-1}$$ (5)

and

$$W_n = \frac{(n-1)\alpha - n}{(n-1)\alpha + 1 - n} W_i$$ (6)

then

$$W_i (n-1) \alpha + 1 - n W_i = \left[ (n-1) \alpha \right]^{-1} \left[ ((n-1) \alpha - n) W_i + 1 \right]$$ (7)

So the optimal value of $W_i$ should satisfy equation (7). When $W_i$ 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 $\text{disp}(W) = \ln n$ which is the optimal solution to (3) for $\alpha = 0.5$.

4. A NEW POWERFUL AND DYNAMIC OWA MODEL

In this section, we introduce the schematic view of proposed model and it’s algorithm.

4.1. A New OWA Aggregation Model

In the opinion of Choi [3], these types of aggregation operators (in [4]) are independent of aggregation situation. Even though Yager’s operator [11] and $\Downarrow$-operator [7] are suggested as an aggregation method using parameter, at present, the definition of such a parameter is still missing [3].

After comparing the operators in [3-4,7,10-11], we found that the OWA operator has the rational aggregation result and near thinking of human being. But the defect is it does not have ability to reflect the aggregative situation during the aggregation process.

For the reason to keep the useful character (rational aggregation result) and correct the shortcomings, we adds two important concepts: (1) modifying the aggregation weights of these attributes “dynamically”, (2) changing these weights “based on situation”. This new model (see Fig. 1) not only has ability to modify forecasting result of function based on the aggregative situation, but also can obtain the associated weights of attributes rely on the OWA operator even match the thinking model of human being.

Fig.1. A schematic view of proposed model

The first concept of modifying the aggregation dynamic weights of attributes is the process that we give the experts who want to evaluate the projects different weights. By this way, they will contribute different
effect to the integral result after evaluation. For instance, if we use the evaluative time as the criterion to measure the degree of information quality, we will give the new-coming expert more weight. This step can let the new-coming expert have more influence on the weights of attributes and evaluation of each project. For this reason, we may get different rating of attributes in respect to the same attribute framework, so that we can naturally obtain dissimilar weights of attributes, and then to make different final proposed solution for decision maker’s reference.

Of therefore will join this function particularly, this is because at manage the group decision in respect to MCDM problems, usually the information of the new-coming has the force of the set of the formerly-coming. Or we can say that in such information century, new information is usually good information. Raising the analysis the stock price the wave motion to an example, perhaps over a hour, have taken place a stock price of impact rises to fall of event, the nature should given the new-coming experts much weight.

Secondly, the concept of changing the weights of each attributes “based on situation” is that the decision maker (or project manager) determine what the value of parameter $\alpha$ is from the information entropy of actual aggregative situation. So, we can use the proposed model to obtain the weights of attributes by the rating of them after OWA aggregation according to $\alpha$. The greatest contribution of this concept is that we can treat our model as a magnifying glass to determine the most important attribute based on the sparest information (i.e. optimistic and $\alpha=0$ or 1) situation (see Table 1’s last column). On the other hand, when $\alpha=0.5$ (moderate situation), our model can get the attributes’ weights (equal attributes’ weights) based on maximal information (such as Table 1’s second column).

4.2. Algorithm of Proposed Model

For easy develop system, we list our proposed model’s algorithm as follows:

Step 1. Build the hierarchical structure model from determining problem and the number of attributes ($N$). (see Fig. 2 $N=6$)

Step 2. Obtain the opinions of the experts in software development, and then use the GDSS to collect their evaluative weights of attributes in respect to the hierarchical structure model of Fig. 2.

Step 3. List the feasible projects, and request the experts to evaluate these projects’ grades in respect to the risk items of Fig. 1.

Step 4. If there is no new-coming expert, execute Step 5. Else if the experts don’t have significant orderings, give the same weight for the evaluation. On the other hand, perform the OWA aggregation process to get the weights for the evaluation, and then execute Step 2. and 3. by this step results.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Risk item</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X_1$ Personnel</td>
<td>$X_5$, Personnel shortcomings, key person(s) quit</td>
</tr>
<tr>
<td>$X_2$ System requirement</td>
<td>$X_6$, Requirement ambiguity</td>
</tr>
<tr>
<td>$X_3$ Schedules and budgets</td>
<td>$X_7$, Developing the wrong software function</td>
</tr>
<tr>
<td>$X_4$ Developing technology</td>
<td>$X_8$, Developing the wrong user interface</td>
</tr>
<tr>
<td>$X_5$ External resource</td>
<td>$X_9$, Continuing iterative requirement changes</td>
</tr>
<tr>
<td>$X_6$ Performance</td>
<td>$X_{10}$, Schedule not accurate</td>
</tr>
<tr>
<td></td>
<td>$X_{11}$, Skills level inadequate</td>
</tr>
<tr>
<td></td>
<td>$X_{12}$, Training hardware</td>
</tr>
<tr>
<td></td>
<td>$X_{13}$, Obtaining software</td>
</tr>
<tr>
<td></td>
<td>$X_{14}$, Shortfall in externally finished component</td>
</tr>
<tr>
<td></td>
<td>$X_{15}$, Shortfall in externally performed tasks</td>
</tr>
<tr>
<td></td>
<td>$X_{16}$, Routine performance shortfall</td>
</tr>
</tbody>
</table>

Fig. 2. Hierarchical structure model of aggregative risk
Step 5. The weights of each expert multiply their evaluative weights of attributes and then use the rating of aggregation result to perform OWA aggregation for getting the refined weights of attributes.

Step 6. Use equation (8) to distribute the weight of each attributes’ risk item(s) based on the ratio of weights of these risk item(s) given by the experts.

$$\bar{W}_{v} = \frac{W_{v}}{\sum_{j} W_{j}}$$

(8)

i=number of attributes; j=number of risk items

Step 7. Multiply the weights of risk items by their grades of the projects, and then rank it’s ordering to make the reference solution to the decision maker.

5. THE OWA GDSS FOR EVALUATING THE RISKS IN SOFTWARE DEVELOPMENT

For evaluating the aggregative risks in software development, we use Borland C++ builder [1] to develop an OWA GDSS (the model is shown as Fig. 1 and 3), and use it to evaluate the attributes of Fig. 2.

The decision makers can use either the importance set $W$={VL, L, M, H, VH} with the grade set $S$ = {DL, EL, VL, L, SL, M, SH, H, VH, EH, DH} or their own preference directly rating by normal triangular fuzzy numbers for accessing the weights of attributes, the weights of risk items, and the grades of risks. After input these experts’ evaluative data into our OWA GDSS, this system will obtain the weights of attributes by the rating of them after OWA aggregation according to $\mu$ and then find out the final proposed solution for the decision maker’s reference. The system’s Output interface is shown as Fig. 3.

6. THE PROPOSED MODEL’S VERIFICATION, COMPARISON AND DISCUSS

In order to verify and compare our model with other methods, we have developed an OWA GDSS for evaluating the risks of software development. We use the data of Lee [6] as input of this GDSS, and then we compare the result with Lee’s algorithm [6] and Chen’s algorithm [13] to validate the accuracy of our proposed model. Simultaneously, we find out the advantages and disadvantages of this GDSS.

6.1 Verifying and Comparing by Numerical Example

In the following, we use the example shown in [6] to illustrate the aggregative risk evaluation process under the group decision making fuzzy environment. We use these data in Lee’s to be the input of our GDSS.

Fig. 3. Output interface of the proposed GDSS
Because the results of our system will be the same with \( \alpha = 0.5 \) and \( \alpha = 0.5 - \delta \), we just need to show the data of \( \alpha = 0.5 \) to stand for the total result. For example, the aggregation results are the same when \( \alpha = 0.7 \) and \( \alpha = 0.3 \). Besides, even if the parameter \( \alpha \) is a continuous value between 0 to 1, we just display the output value 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 data in [6], we summarize the output weights of the attributes in Table 1. From Table 1’s last column, we can see our model as a magnifying glass to determine the most important attribute based on the sparsest information (i.e., optimistic and \( \alpha = 0 \) or 1) situation. In Table 1’s second column, when \( \alpha = 0.5 \) (moderate situation), our model can get the attributes’ weights (equal attributes’ weights) based on maximal information.

Similarly, we take the \( \alpha \) value from 0.5 to 1.0 as the parameter to execute according to our algorithm for the purpose of verification in respect to proposed model and the result is shown below as Table 2.

For the further reason to verify the validity of this proposed model, we compare the result of our algorithm with the algorithms in Lee [6] and Chen [13]. But, our aggregation result will change corresponding to the \( \alpha \) value, so we choose the extreme values of \( \alpha \) as representation. Therefore, in Table 3., we select two extreme points of maximum (\( \alpha = 0.5 \)) and minimum information (\( \alpha = 1 \) or 0) entropy to summarize the aggregation result of the three projects.

From Table 2., we can see the rank of the proposed algorithm is that the risk of Project (\( \alpha = 0.5 \)) > Project (\( \alpha = 0.6 \)) > Project (\( \alpha = 0.7 \)), which is the same as the ratings of the projects calculated by Lee’s and Chen’s algorithm. So, we can say that our proposed model is validated.

### 6.2 Results and Discuss

**Result 1:** After the aggregation results by different \( \alpha \)’s comparison, the values of risks calculated by Lee’s and Chen’s algorithm are less than our algorithm even if \( \alpha = 0.5 \) or \( \alpha = 1.0 \) (or 0).

This is because that they use the matrix multiplication of weights and grades in their approaches and their algorithms aren’t normalized the weights of attributes when calculation. In contrast with our model, to be normalized is the characteristic of OWA operators and equation (18) can also contribute to the normalization of the risk items during aggregation process.

**Result 2:** From Table 1, if the \( \alpha \) value changes from 0.5 to 1.0 (or 0), the weights of attributes will also be changed from distributed (equal weights) to centralized (the most important attribute).

### Table 1. The weights of attributes after OWA aggregation

<table>
<thead>
<tr>
<th>Attribute</th>
<th>( \alpha = 0.5 )</th>
<th>( \alpha = 0.6 )</th>
<th>( \alpha = 0.7 )</th>
<th>( \alpha = 0.8 )</th>
<th>( \alpha = 0.9 )</th>
<th>( \alpha = 1.0 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Personnel (W₁)</td>
<td>0.1666</td>
<td>0.1461</td>
<td>0.1142</td>
<td>0.0723</td>
<td>0.0255</td>
<td>0</td>
</tr>
<tr>
<td>System requirement (W₂)</td>
<td>0.1666</td>
<td>0.2468</td>
<td>0.3475</td>
<td>0.4781</td>
<td>0.6637</td>
<td>1</td>
</tr>
<tr>
<td>Schedules and budgets (W₃)</td>
<td>0.1666</td>
<td>0.1031</td>
<td>0.0544</td>
<td>0.0205</td>
<td>0.0029</td>
<td>0</td>
</tr>
<tr>
<td>Developing technology (W₄)</td>
<td>0.1666</td>
<td>0.2072</td>
<td>0.2398</td>
<td>0.2547</td>
<td>0.2240</td>
<td>0</td>
</tr>
<tr>
<td>External resource (W₅)</td>
<td>0.1666</td>
<td>0.1227</td>
<td>0.0788</td>
<td>0.0385</td>
<td>0.0086</td>
<td>0</td>
</tr>
<tr>
<td>Performance (W₆)</td>
<td>0.1666</td>
<td>0.1740</td>
<td>0.1654</td>
<td>0.1357</td>
<td>0.0756</td>
<td>0</td>
</tr>
</tbody>
</table>

**Table 2. The aggregation result of our algorithm**

<table>
<thead>
<tr>
<th>Project</th>
<th>( \alpha = 0.5 )</th>
<th>( \alpha = 0.6 )</th>
<th>( \alpha = 0.7 )</th>
<th>( \alpha = 0.8 )</th>
<th>( \alpha = 0.9 )</th>
<th>( \alpha = 1.0 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project I</td>
<td>0.4260</td>
<td>0.4197</td>
<td>0.4110</td>
<td>0.3997</td>
<td>0.3873</td>
<td>0.3867</td>
</tr>
<tr>
<td>Project II</td>
<td>0.7102</td>
<td>0.7136</td>
<td>0.7185</td>
<td>0.7253</td>
<td>0.7355</td>
<td>0.7455</td>
</tr>
<tr>
<td>Project III</td>
<td>0.2033</td>
<td>0.2025</td>
<td>0.2021</td>
<td>0.2010</td>
<td>0.1966</td>
<td>0.1773</td>
</tr>
</tbody>
</table>

**Table 3. The comparison of the previous and our proposed algorithm**

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Project</th>
<th>( \alpha = 0.5 )</th>
<th>( \alpha = 0.6 )</th>
<th>( \alpha = 0.7 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Algorithm 1 by Lee [6]</td>
<td>0.19650</td>
<td>0.50917</td>
<td>0.09112</td>
<td></td>
</tr>
<tr>
<td>Algorithm 2 by Lee [6]</td>
<td>0.19648</td>
<td>0.50930</td>
<td>0.09173</td>
<td></td>
</tr>
<tr>
<td>Algorithm by Chen [13]</td>
<td>0.18541</td>
<td>0.51086</td>
<td>0.05767</td>
<td></td>
</tr>
<tr>
<td>Our algorithm (( \alpha = 0.5 ))</td>
<td>0.42598</td>
<td>0.71018</td>
<td>0.20332</td>
<td></td>
</tr>
<tr>
<td>Our algorithm (( \alpha = 0 ) or 0)</td>
<td>0.38668</td>
<td>0.74545</td>
<td>0.17727</td>
<td></td>
</tr>
</tbody>
</table>

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 GDSS to search out the critical attribute of the problem when the $\alpha$ value given by the project manager is 0 or 1 (i.e. under the minimal entropy). Therefore, this system would be a useful tool if the project manager wants to find the most important attribute (or criterion).

7. CONCLUSION

In this paper, we proposed a new powerful and dynamic OWA model to deal with the fuzzy MCDM problems and verify its validity by an example of aggregative risks in software development based on a computer-based GDSS. The advantages of this new model are:

1. Using the fuzzy linguistic variables to help the decision maker to get the weight of criteria faster and more reasonable (based on situation).

2. It can work like a magnifying glass to adjust its focus based on the sparsest information (i.e. optimistic situation $\alpha=0$ or 1) to find the most important attribute, or based on maximal information (i.e. moderate situation $\alpha=0.5$) to get equal attributes' weights.

3. Using the GDSS based on our model to solve the MCDM problem more efficient.

4. Help user (project manager) can easily maintain the scale of measure (see Result 1).

Mendel [8] suggests use five (or seven) scales when facing the fuzzy MCDM problems represented by linguistic variables, in the future, we may change the scale of fuzzy linguistic variables from eleven to five (or seven). Nowadays, the applications of fuzzy neuron network (FNN) become a better solution to the uncertain problem. For this reason, we can combine the fuzzy rule base of FNN with our new powerful and dynamic OWA model so that this model can help the fuzzy rule base to converge faster. Besides, in our opinion, the proposed model would be a useful tool in dealing with every kind of MCDM problems for the decision makers, such as performance evaluation, public policy, the resource of the aqua and the subject of the environment etc.

REFERENCE