

# We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

**4,800**

Open access books available

**122,000**

International authors and editors

**135M**

Downloads

Our authors are among the

**154**

Countries delivered to

**TOP 1%**

most cited scientists

**12.2%**

Contributors from top 500 universities



**WEB OF SCIENCE™**

Selection of our books indexed in the Book Citation Index  
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?  
Contact [book.department@intechopen.com](mailto:book.department@intechopen.com)

Numbers displayed above are based on latest data collected.

For more information visit [www.intechopen.com](http://www.intechopen.com)



---

# Fuzzy Analytical Network Process Implementation with Matlab

---

Xiaoguang Zhou

Additional information is available at the end of the chapter

<http://dx.doi.org/10.5772/46466>

---

## 1. Introduction

In many complex decision-making problems, the decision information provided by the decision makers is often imprecise or uncertain due to time limit, lack of data, or the decision makers' limited attention and information processing capabilities. Decision-making in a fuzzy environment means a decision process in which the goals and/or the constraints, but not necessarily the system under control, are fuzzy in nature. This means that the goals and/or the constraints constitute classes of alternatives whose boundaries are not sharply defined. Fuzzy set, whose basic component is a membership function (Zadeh, 1965), was introduced in the following several decades. Fuzzy set theory has been applied successfully in the decision-making field.

Matlab is a suitable tool for solving fuzzy decision-making problems. This chapter is focusing on how to solve a specific class of fuzzy decision-making problem, that is, Fuzzy Analytical Network Process (FANP) by Matlab. Project selection is chosen as an example to illustrate the proposed method. The reason is that project selection is a complex decision-making process. It involves a search from the environment of opportunities, the generation of project options, and the evaluation of multiple attributes, both qualitative and quantitative, by different stakeholders.

There are various mathematic techniques for selecting an optimal project. Mathematical programming models can be used to accomplish this decision. For example, the R&D project selection can be presented based on linear, non-linear, dynamic, goal, and stochastic mathematical programming (Wang & Hwang, 2007). Based on the idea of moments approximation method via linear programming, Fang et al. (2008) proposed a scenario generation approach for the mixed single-stage Research and Development (R&D) projects and multi-stage securities portfolio selection problem, etc.

Also, project selection has been presented in regard with multiple objectives. For instance, Gabriel et al. (2006) developed a multiobjective, integer-constrained optimization model with competing objectives for project selection, and the subjective rank is determined via the Analytic Hierarchy Process (AHP). Ghorbani & Rabbani (2009) proposed a multi-objective algorithm for project selection. Two objective functions have been considered to maximize total expected benefit of selected projects and minimize the summation of the absolute variation of allotted resource between each successive time periods. Gutjahr et al. (2010) presented a multi-objective optimization model for project portfolio selection taking employee competencies and their evolution into account, and so on.

Fuzzy sets theory is utilized to cover the vagueness inherent in the nature of project selection problem as well. For example, Huang et al. (2008) presented a fuzzy analytic hierarchy process method and utilize crisp judgment matrix to evaluate subjective expert judgments. Bhattacharyya et al. (2011) developed a fuzzy multi-objective programming approach to facilitate decision making in the selection of R&D projects. Ebrahimnejad et al. (2011) considered a two-phase decision making approach, combining a modified version of the Analytic Network Process (ANP) method and an improved compromise ranking method under uncertainty. Chang & Lee (2012) proposed a Data Envelopment Analysis (DEA), knapsack formulation and fuzzy set theory integrated model to deal with the project selection, etc.

Some researchers tried the project selection problem in other ways. Dey (2006) proposed a decision support system, which analyses projects with respect to market, technicalities, and social and environmental impact in an integrated framework using analytic hierarchy process. Liesiö et al. (2007) developed the Robust Portfolio Modeling methodology which extends Preference Programming methods into portfolio problems, where a subset of project proposals are funded in view of multiple evaluation criteria. Smith-Perera et al. (2010) proposed an approach to prioritize project portfolio in an efficient and reliable way based on analytic network process method and the information obtained from the experts during the decision-making process. Shakhshi-Niaei et al. (2011) presented a procedure which used the PROMETHEE method linked to a Monte Carlo simulation in order to consider and possibly make lower all kinds of uncertainties of project selection problem in an acceptable complexity level, and so on.

As mentioned above, AHP or ANP was widely used during the process of selecting or evaluating an optimal project (Gabriel et al., 2006; Dey, 2006; Smith-Perera et al., 2010; Ebrahimnejad et al., 2011), and these literatures (Cheng & Li, 2005; Amiri, 2010; Aragonés-Beltrán et al., 2010; Jung & Seo, 2010; etc) are also about project selection based on AHP/ANP. In addition, AHP/ANP was proverbially used in other decision-making fields as well (Kahraman et al., 2006; Lee et al., 2009; Arunraj & Maiti, 2010; Huang et al., 2011; etc.). The main reason is that AHP/ANP can deal with qualitative and quantitative information at the same time, and ANP can take into account the interaction and feedback relationships between criteria and/or indices. However, due to the vagueness and uncertainty on the judgments of decision-makers, the crisp pairwise comparison in the conventional AHP/ANP seems insufficient and imprecise to capture the right judgments of decision-makers.

In this study, a fuzzy logic is introduced for the pairwise comparison of ANP to make up the deficiency of the conventional AHP/ANP, referred to as FANP. The objective of this chapter is to present a FANP-based approach for the construction project selection problem using triangular fuzzy numbers. According to the fuzzy preference programming (FPP) method, local weights of fuzzy pairwise comparison matrices can be achieved. Then an unweighted and weighted supermatrix based on its network structure can be formed. For FANP, the key steps are to calculate the local weights and the limit supermatrix. Both of them can be solved by Matlab. A case will be given by the proposed method, and Matlab codes will be provided as well.

## 2. What's FANP?

The ANP, introduced by Saaty, is a generalization of the AHP (Saaty, 1996). ANP is the first mathematical theory that makes it possible to deal with all kinds of dependences and feedbacks by replacing hierarchies with networks (Saaty, 1996). ANP allows for complex inter-relationships among decision dimensions and attributes. The ANP feedback approach replaces hierarchies with networks in which the relationships between dimensions are not easily represented as higher or lower, dominant or subordinate, direct or indirect. For instance, not only does the importance of the criteria determine the importance of the attributes, as in a hierarchy, but also the importance of the attributes may have impact on the importance of the criteria. A hierarchical structure with a linear top-to-bottom form is not suitable for a complex system.

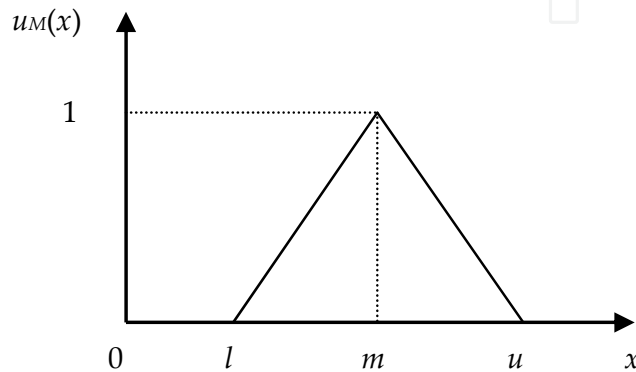
As we know, AHP/ANP has been proposed as a suitable multi-criteria decision analysis tool for project selection and evaluation. However, the conventional AHP/ANP-based decision model seems to be ineffective in dealing with the inherent fuzziness or uncertainty in judgment during the pairwise comparison process. Although the use of the discrete scale of 1-9 to represent the verbal judgment in pairwise comparisons has the advantage of simplicity, it does not take into account the uncertainty associated with the mapping of one's perception or judgment to a number. In real-life decision-making situations, the decision makers or stakeholders could be uncertain about their own level of preference, due to incomplete information or knowledge, complexity and uncertainty within the decision environment. Such situations will occur when selecting and evaluating an optimal project. Therefore, it's better to make project selection and assessment under fuzzy conditions. This chapter will focus on FANP in fuzzy decision-making based on triangular fuzzy numbers. Actually, some researchers have focused on decision-making based on FANP (Promentilla et al., 2008 ; Ayağ & Özdemir, 2009 ; Boran & Goztepe, 2010; Dağdeviren & Yüksel, 2010; Pires et al., 2011 ; Ju et al., 2012 ; etc. ).

### 2.1. Triangular fuzzy number

A fuzzy set is a class of objects with a continuum of grades of membership. Such a set is characterized by a membership function, which assigns to each object a grade of membership ranging between zero and one.

A triangular fuzzy number (TFN)  $M$  is shown in Fig. 1. A TFN is denoted simply as  $(l, m, u)$ . The parameters  $l$ ,  $m$  and  $u$ , respectively, denote the smallest possible value, the most promising value, and the largest possible value that describe a fuzzy event. Each TFN has linear representations on its left and right side such that its membership function can be defined as

$$u_M(x) = \begin{cases} (x-l)/(m-l) & l \leq x \leq m \\ (u-x)/(u-m) & m \leq x \leq u \\ 0 & \text{otherwise} \end{cases} \quad (1)$$



**Figure 1.** A triangular fuzzy number  $M$

## 2.2. Fuzzy preference programming method

A number of methods have been developed to handle fuzzy comparison matrices. For instance, Van Laarhoven & Pedrycz (1983) suggested a fuzzy logarithmic least squares method to obtain the fuzzy weights from a triangular fuzzy comparison matrix. Buckley (1985) utilized the geometric mean method to calculate fuzzy weights. Chang (1996) proposed an extent analysis method, which derives crisp weights for fuzzy comparison matrices. Xu (2000) brought forward a fuzzy least squares priority method (LSM). Csutora & Buckley (2001) came up with a Lambda-Max method, which is the direct fuzzification of the well-known  $k$ max method. Mikhailov (2003, 2004) developed a fuzzy preference programming method, which also derives crisp weights from fuzzy comparison matrices. Srdjevic (2005) proposed a multicriteria approach for combining prioritization methods within the AHP, including additive normalization, eigenvector, weighted least-squares, logarithmic least-squares, logarithmic goal programming and fuzzy preference programming. Wang et al. (2006) presented a modified fuzzy logarithmic least square method. Yu & Cheng (2007) developed a multiple objective programming approach for the ANP to obtain all local priorities for crisp or interval judgments at one time, even in an inconsistent situation. Huo et al. (2011) proposed new parametric prioritization methods (PPMs) to determine a family of priority vectors in AHP.

FPP method, as a reasonable and effective means, is adopted in this study. This method can acquire the consistency ratios of fuzzy pairwise comparison matrices without additional

study, and the local weights can be easily solved by Matlab software. The stages of Mikhailov’s fuzzy prioritization approach are given as follows (Mikhailov, 2003).

Consider a prioritization problem with  $n$  elements, where the pairwise comparison judgements are represented by normal fuzzy sets or fuzzy numbers. Supposing that the decision-maker can provide a set  $F=\{\tilde{a}_{ij}\}$  of  $m \leq n(n-1)/2$  fuzzy comparison judgements,  $i=1, 2, \dots, n-1; j=2, 3, \dots, n; j>i$ , represented as triangular fuzzy numbers  $\tilde{a}_{ij}=(l_{ij}, m_{ij}, u_{ij})$ . The problem is to derive a crisp priority vector  $w=(w_1, w_2, \dots, w_n)^T$ , such that the priority ratios  $w_i/w_j$  are approximately within the scopes of the initial fuzzy judgments, or

$$l_{ij} \lesssim \frac{w_i}{w_j} \lesssim u_{ij} \tag{2}$$

where the symbol  $\lesssim$  denotes the statement “fuzzy less or equal to”.

Each crisp priority vector  $w$  satisfies the double-side inequality (2) with some degree, which can be measured by a membership function, linear with respect to the unknown ratio  $w_i/w_j$

$$u_{ij}\left(\frac{w_i}{w_j}\right) = \begin{cases} \frac{(w_i/w_j) - l_{ij}}{m_{ij} - l_{ij}}, & \frac{w_i}{w_j} \leq m_{ij}, \\ \frac{u_{ij} - (w_i/w_j)}{u_{ij} - m_{ij}}, & \frac{w_i}{w_j} \geq m_{ij}. \end{cases} \tag{3}$$

The membership function (3) is linearly increasing over the interval  $(-\infty, m_{ij})$  and linearly decreasing over the interval  $(m_{ij}, \infty)$ . The function takes negative values when  $w_i/w_j < l_{ij}$  or  $w_i/w_j > u_{ij}$ , and has a maximum value  $u_{ij}=1$  at  $w_i/w_j=m_{ij}$ . Over the range  $(l_{ij}, u_{ij})$ , the membership function (3) coincides with the fuzzy triangular judgment  $(l_{ij}, m_{ij}, u_{ij})$ .

The solution to the prioritization problem by the FPP method is based on two main assumptions. The first one requires the existence of non-empty fuzzy feasible area  $P$  on the  $(n-1)$  dimensional simplex  $Q^{n-1}$

$$Q^{n-1} = \{(w_1, w_2, \dots, w_n) \mid w_i > 0, \sum_{i=1}^n w_i = 1\}, \tag{4}$$

defined as an intersection of the membership functions, similar to (3) and the simplex hyperplane (4). The membership function of the fuzzy feasible area is given by

$$u_p(w) = \min_{ij} \{u_{ij}(w) \mid i = 1, 2, \dots, n-1; j = 2, 3, \dots, n; j > i\}. \tag{5}$$

If the fuzzy judgements are very inconsistent, then  $u_p(w)$  could take negative values for all normalized priority vectors  $w \in Q^{n-1}$ .

The second assumption of the FPP method specifies a selection rule, which determines a priority vector, having the highest degree of membership in the aggregated membership

function (5). It can easily be proved that  $u_p(w)$  is a convex set, so there is always a priority vector  $w^* \in Q^{n-1}$  that has a maximum degree of membership

$$\lambda^* = u_p(w^*) = \max_{w \in Q^{n-1}} \min_{ij} \{u_{ij}(w)\}. \quad (6)$$

The maximum prioritization problem (6) can be represented in the following way:

$$\begin{aligned} & \text{Max } \lambda \\ & \lambda \leq u_{ij}(w), i = 1, 2, \dots, n-1; j = 2, 3, \dots, n; j > i, \\ & \sum_{k=1}^n w_k = 1, w_k > 0, k = 1, 2, \dots, n. \end{aligned} \quad (7)$$

Taking the specific form of the membership functions (3) into consideration, the problem (7) can be further transformed into a bilinear program of the type

$$\begin{aligned} & \text{Max } \lambda \\ & (m_{ij} - l_{ij})\lambda w_j - w_i + l_{ij}w_j \leq 0, \\ & (u_{ij} - m_{ij})\lambda w_j + w_i - u_{ij}w_j \leq 0, \\ & \sum_{k=1}^n w_k = 1, w_k > 0, k = 1, 2, \dots, n. \\ & i = 1, 2, \dots, n-1; j = 2, 3, \dots, n; j > i. \end{aligned} \quad (8)$$

The optimal solution to the non-linear problem above ( $w^*$ ,  $\lambda^*$ ) might be obtained by employing some appropriate numerical method for non-linear optimization. The optimal value  $\lambda^*$ , if it is positive (the maximum value is one), indicates that all solution ratios satisfy the fuzzy judgment completely, which means that the initial set of fuzzy judgments is rather consistent. A negative value of  $\lambda^*$  shows that the solutions ratios approximately satisfy all double-side inequalities (2). Therefore, the optimal value  $\lambda^*$  can be used for measuring the consistency of the initial set of fuzzy judgments.

### 3. Proposed project selection and evaluation framework

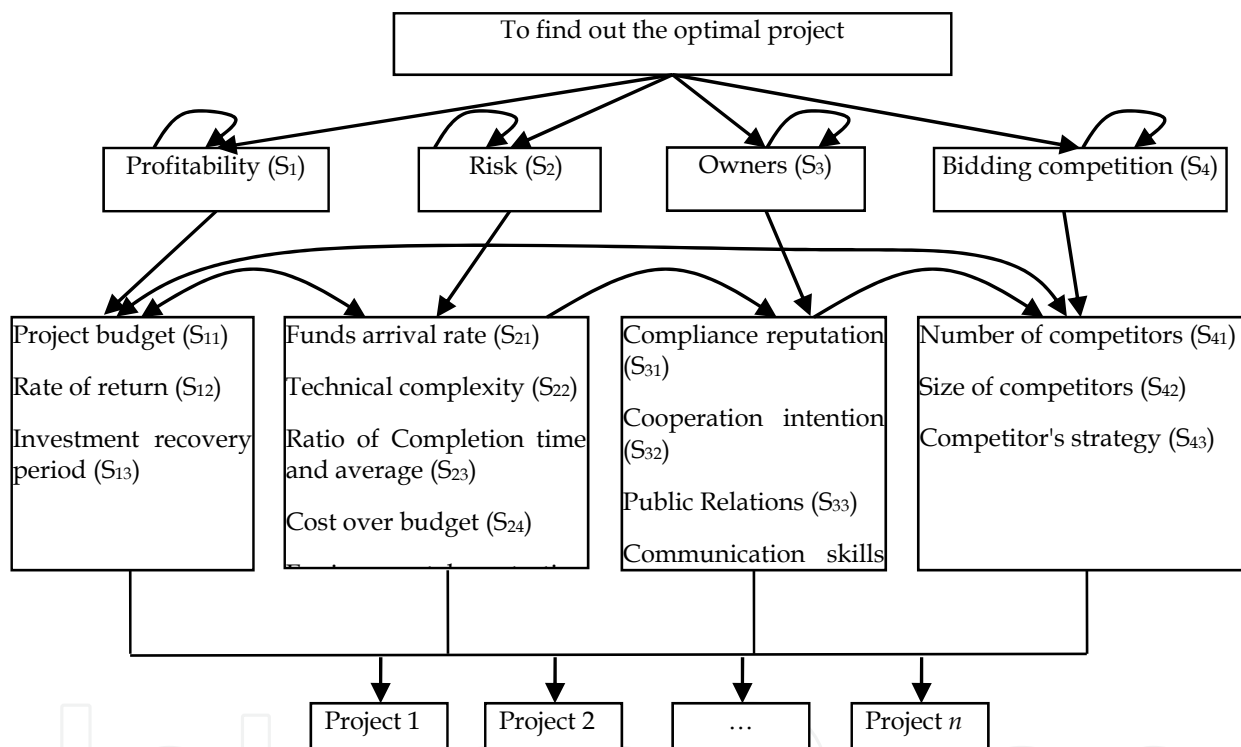
This study proposes an analytic approach based on the fuzzy ANP to assist in project selection and evaluation. We first identify the selection and evaluation criteria. Then we present the evaluation model in the following subsections.

#### 3.1. Criteria of project selection

Some researchers investigated which factors have an effect on project selection. For example, Mohanty (1992) pointed out that a potential project should have four important features: minimum investment, low complex of technology, short period and high returns. Lin & Chen (2004) suggested that the contractors should consider the essence of bidding, competition, the value of tendering opportunities, resources, and corporate reputation.

Kumar (2006) pointed out that project selection belongs to the strategic level for any organizations, and today it is seriously influenced by environmental and social factors. Wang et al. (2009) suggested that during project selection, the evaluation index system can be divided into two parts: the bid/non-bid and which project to bid, then make a comprehensive evaluation, etc.

With the interaction and feedback relationships between dimensions and/or attribute-enablers being considered, a four-level evaluation index system is presented, as shown in Fig. 2. The first level is the objective, which is to find out an optimal project; the second level is the dimensions including project profitability, project risk, project owners and project bidding competition; the third level is attribute-enablers, 15 indicators included; and the lowest level is the alternatives.



**Figure 2.** The ANP structure for construction project selection

Generally, if project profitability ( $S_1$ ) has an effect on project bidding competition ( $S_4$ ), then a line with arrow from  $S_4$  to  $S_1$  is added. If the sub-criteria of project risk ( $S_2$ ) has interaction with itself, then  $S_2$  is inner dependent, and an arc with arrow is added to  $S_2$ .

### 3.2. FANP-based approach

The fuzzy ANP-based approach is presented step-by-step as follows:

**Step 1.** Model construction and problem structuring. With the relationships among dimensions and attribute-enablers being considered, a four levels selection and evaluation index system is proposed, as shown in Fig. 2.



During the process of project selection and evaluation, experts tend to specify their preferences in the form of natural language expressions. The fuzzy linguistic variable, whose value represents the range from natural to artificial language, is a variable that reflects different aspects of human language. When the values or meanings of a linguistic factor are being reflected, the resulting variable must also reflect appropriate modes of change for that linguistic factor. Moreover, variables describing a human word or sentence can be divided into numerous linguistic criteria, such as equally important, moderately important, important, very important or absolutely important. For the purposes of the present study, a 9-point scale is presented for the relative importance of pairwise comparison, as shown in Table 1.

**Step 2.** Establishing pairwise comparison matrices by decision committee using the linguistic scales for relative importance given in Table 1. For example, complication of technique ( $S_{11}$ ) and maturity of technique ( $S_{12}$ ) are compared using the question “How important is the complication of technique when it is compared with the maturity of technique at the dimension of technique risk?” and the answer is “moderately important (MI)”, so this linguistic scale is placed in the relevant cell against the triangular fuzzy numbers (2, 3, 4). All the fuzzy evaluation matrices are produced in the same manner.

Linguistic scale for importance	Triangular fuzzy scale	Triangular fuzzy reciprocal scale
Equally important(EI)	(1, 1, 1)	(1, 1, 1)
Intermediate1(IM <sub>1</sub> )	(1, 2, 3)	(1/3, 1/2, 1)
Moderately important(MI)	(2, 3, 4)	(1/4, 1/3, 1/2)
Intermediate2(IM <sub>2</sub> )	(3, 4, 5)	(1/5, 1/4, 1/3)
Important(I)	(4, 5, 6)	(1/6, 1/5, 1/4)
Intermediate3(IM <sub>3</sub> )	(5, 6, 7)	(1/7, 1/6, 1/5)
Very important(VI)	(6, 7, 8)	(1/8, 1/7, 1/6)
Intermediate4(IM <sub>4</sub> )	(7, 8, 9)	(1/9, 1/8, 1/7)
Absolutely important(AI)	(9, 9, 9)	(1/9, 1/9, 1/9)

**Table 1.** Linguistic scales for relative importance

**Step 3.** Calculating the local weights of the factors and sub-factors taking part in the second and third levels of the ANP model by FPP method according to formulation (8).

**Step 4.** Constructing an unweighted supermatrix based on the interdependencies in the network. The supermatrix is a partitioned matrix, in which each submatrix is composed of a set of relationships between dimensions and attribute-enablers in the graphical model.

**Step 5.** Acquiring the weighted supermatrix. Because in each column it consists of several eigenvectors which of them sums to one (in a column of a stochastic), and hence the entire column of the matrix may sum to an integer greater than one, the unweighted supermatrix needs to be stochastic to derive the weighted supermatrix.

**Step 6.** Obtaining the limit supermatrix. The weighted supermatrix will not be in a steady state until the row values of which converge to the same value for each column of the matrix, then the limit supermatrix is achieved.

$$\overline{W} = \lim_{t \rightarrow \infty} W^t. \tag{9}$$

**Step 7.** Calculating the comprehensive weights. A comprehensive weight of each index can be obtained by multiplying the weight of the criterion level indicator, the weight of independent sub-criterion and the weight of interdependent sub-index.

$$w_{ij} = P_i \times A_{ij}^D \times A_{ij}^I, \tag{10}$$

where  $w_{ij}$  is the comprehensive weight of each factor,  $P_i$  is relative importance weight of dimension  $i$  on final goal;  $A_{ij}^D$ , relative importance weight for attribute-enabler  $j$  of dimension  $i$ , and for the dependency ( $D$ ) relationships within attribute-enabler's component level;  $A_{ij}^I$ , stabilized relative importance weight for attribute-enabler  $j$  of dimension  $i$ , and for the independency ( $I$ ) relationships within attribute-enabler's component level;  $i=1, 2, \dots, m; j=1, 2, \dots, n$ .

**Step 8.** Selecting an optimal construction project. The equation of desirability index,  $D_k$  for alternative  $k$  is calculated using the following equation:

$$D_k = \sum_{i=1}^m \sum_{j=1}^n w_{ij} S_{kij} \tag{11}$$

where  $w_{ij}$  is the comprehensive weight;  $S_{kij}$ , relative impact of construction project alternative  $k$  on attribute-enabler  $j$  of dimension  $i$  of selection network.

#### 4. How to use Matlab during the process of decision-making with FANP?

Two key steps in the process of FANP will be solved by Matlab. One is acquiring local weights of fuzzy pairwise comparison matrices (step 3); the other is obtaining the limit supermatrix (step 6). Both of them are associated with matrix calculation. Matlab is selected for its excellent performance on matrix operation and data processing.

##### 4.1. Acquiring local weights of fuzzy pairwise comparison matrices

$$\begin{aligned} & \text{Max } \lambda \\ & (m_{ij} - l_{ij})\lambda w_j - w_i + l_{ij}w_j \leq 0, \\ & (u_{ij} - m_{ij})\lambda w_j + w_i - u_{ij}w_j \leq 0, \\ & \sum_{k=1}^n w_k = 1, w_k > 0, k = 1, 2, \dots, n. \\ & i = 1, 2, \dots, n - 1; j = 2, 3, \dots, n; j > i. \end{aligned}$$

As mentioned before, the local weights of fuzzy pairwise comparison matrices are achieved by FPP method. That is, the local weights will be obtained by solving the non-linear problem above.

As criteria and sub-criteria have different numbers, there are different orders of fuzzy pairwise comparison matrices, such as matrices of order  $(2 \times 2)$ ,  $(3 \times 3)$ , ...,  $(n \times n)$ . Therefore, the local weights and consistency index may be derived from matrix of different orders. Function definition and non-linear program calculation will be first concerned in this section, then some examples will be given to illustrate the proposed method.

#### 4.1.1. Function definition

To acquire the local weights of fuzzy comparison matrices, some functions need to be defined. A procedure can be saved as the format - ".m" file in Matlab. The name of which can be used directly when you need to call it.

To obtain the local weights, a main program file is developed, named as "networkmain.m". The local weights can be easily acquired by inputting "networkmain" in the command window. As there are different forms of comparison matrices, we need to change slightly the main program file for matrices of different orders.

Matlab contains a lot of functions to solve linear and non-linear problems. For instance, non-linear problems can be solved by function "fmincon", which is also the key function of the file "networkmain.m". The full expression of the function is "fmincon (fun, x0, A, b, Aeq, beq, VLB, VUB, nonlcon)". During the process of decision-making with FANP, in accordance with the FPP method, the fuzzy pairwise comparison matrices first need to be transformed into non-linear programming formats. Then it can be solved by the function "fmincon". The detail description of the function is as follows:

"Fun" is the objective function of a non-linear problem. A variable in the objective function is marked as  $x(i)$ . If the total number of variables is  $n$ , then they are correspondingly named as  $x(1)$ ,  $x(2)$ , ...,  $x(n)$ . There are  $(n+1)$  variables for a  $(n \times n)$  comparison matrix, including  $n$  local weights and a consistency index  $\lambda$ . The objective function in FPP method is to acquire the maximum value, but the default standard objective function of "fmincon" in Matlab is to find the minimum value, so it is necessary to convert  $x(n+1)$  into  $-x(n+1)$  in the function "fmincon". Of course, it might be solved by changing the constraints instead of changing the objective function as well.

"x0" is the initial value of the non-linear problem, and it has the same scale as the number of variables. Every local weight  $x(i)$  takes values in the range  $[0, 1]$ , and their sum satisfies  $x(1)+x(2)+\dots+x(n)=1$ . Consistency index  $x(n+1)$  takes values in the range  $(-\infty, 1]$ .

"A and b" are the coefficients of linear inequality constraint  $Ax \leq b$ . As there are no linear inequality constraints in FANP,  $a$  and  $b$  can be ignored or replaced with two empty arrays.

"Aeq and beq" are both the coefficients of linear equality constraint  $Aeq \cdot x = beq$ . According to FANP, the sum of local weights should be one, then

$$x(1) + x(2) + \dots + x(n) + 0x(n+1) = 1,$$

where  $x(1), x(2), \dots, x(n)$  are the first, second, ...,  $n$ th local weight respectively, and  $x(n+1)$  is the consistency index. Then we have  $A_{eq}=[1 \ 1 \ \dots \ 1 \ 0]$  and  $b_{eq}=[1]$ .

“VLB and VUB” are the upper and lower bounds of the variables. According to the FPP method and FANP, all the local weights have a lower bound of zero, and the lower bound of consistency index is negative infinity. Since all the upper bounds are subject to the constraints, they can be replaced with empty arrays. In matlab, the positive and negative infinity symbols are named as `inf` and `-inf` respectively.

“nonlcon” is the non-linear constraints, including non-linear inequality constraint  $c$  and non-linear equality constraint  $ceq$ . As there is no non-linear equality constraint for FPP method, we can let  $ceq=[]$ .

To solve the non-linear problem (8), the following main program file “networkmain.m” is developed.

```
Aeq=[1 1 ... 1 0];
beq=[1];
VLB = [0; 0; ...; 0; -inf];
VUB = [ ];
x0 = [0.1; 0.2; ...; 1];
OPT = optimset('LargeScale', 'off');
[x, fval] = fmincon('networkf', x0, [ ], [ ], Aeq, beq, VLB, VUB, 'networknonlcon', OPT)
```

The `LargeScale` option specifies a preference for which algorithm to use. It is only a preference because certain conditions must be met to use the large-scale algorithm. For this function “fmincon”, we choose the medium-scale algorithm. `LargeScale` use the medium-scale algorithm when set to 'off'. Two files, “networkf.m” and “networknonlcon.m”, are called in the procedure above. They are defined for objective function and non-linear constraints independently.

The program of “networkf.m” is developed as follows:

```
function f=networkf(x);
f=-x(n+1);
```

where the value of function  $f$  is related to  $n$ . For example, for a matrix of order 3,  $n=3$ ,  $f=-x(4)$ ; for a matrix of order 4,  $n=4$ ,  $f=-x(5)$ . Therefore, objective function  $f$  has different formats as a result of different sizes of matrices.

According to the FPP method, every triangular fuzzy number  $(l_{ij}, m_{ij}, u_{ij})$  needs to be transformed into the following inequality constraints.

$$(m_{ij} - l_{ij}) * x(n+1) * x(j) - x(i) + (l_{ij}) * x(j) \leq 0;$$

$$(u_{ij} - m_{ij}) * x(n+1) * x(j) + x(i) - (u_{ij}) * x(j) \leq 0.$$

As we know, a triangular fuzzy comparison matrix is symmetric. Therefore, for the following matrix, we only need to consider the constraints above the diagonal.

$$\begin{pmatrix} (l_{11}, m_{11}, u_{11}) & (l_{12}, m_{12}, u_{12}) & \dots & (l_{1n}, m_{1n}, u_{1n}) \\ (l_{21}, m_{21}, u_{21}) & (l_{22}, m_{22}, u_{22}) & \dots & (l_{2n}, m_{2n}, u_{2n}) \\ \vdots & \vdots & \vdots & \vdots \\ (l_{n1}, m_{n1}, u_{n1}) & (l_{n2}, m_{n2}, u_{n2}) & \dots & (l_{nn}, m_{nn}, u_{nn}) \end{pmatrix}$$

For instance, for a 3-by-3 fuzzy comparison matrix, only three elements need to be taken into account, which is,  $(l_{12}, m_{12}, u_{12})$ ,  $(l_{13}, m_{13}, u_{13})$  and  $(l_{23}, m_{23}, u_{23})$ . Then, the corresponding “networknonlcon.m” file is given by

function [c, ceq] = networknonlcon3(x);

```
c = [
    (m12-l12)*x(4)*x(2)-x(1)+(l12)*x(2);
    (u12-m12)*x(4)*x(2)+x(1)-(u12)*x(2);
    (m13-l13)*x(4)*x(3)-x(1)+(l13)*x(3);
    (u13-m13)*x(4)*x(3)+x(1)-(u13)*x(3);
    (m23-l23)*x(4)*x(3)-x(2)+(l23)*x(3);
    (u23-m23)*x(4)*x(3)+x(2)-(u23)*x(3);
];
ceq = [ ];
```

The programs for  $(4 \times 4)$ ,  $(5 \times 5)$ , ...,  $(n \times n)$  matrices can be developed in the same manner. Several examples for acquiring the local weights are given as follows.

#### 4.1.2. Local weights of a fuzzy pairwise comparison matrix of order 3

**Example 1.** How to solve the local weights of the following fuzzy pairwise comparison matrix of order 3?

$$\begin{pmatrix} (1,1,1) & (1/6,1/5,1/4) & (1/4,1/3,1/2) \\ (4,5,6) & (1,1,1) & (2,3,4) \\ (2,3,4) & (1/4,1/3,1/2) & (1,1,1) \end{pmatrix}$$

First, the initial parameter of the main function “networkmain” needs to be set for this  $(3 \times 3)$  matrix, which has three local weights, named as  $x(1)$ ,  $x(2)$  and  $x(3)$ , and a consistency index, referred to as  $x(4)$ . That means there are four variables in linear equality constraint. The corresponding “networkmain.m” file is as follows:

```
Aeq=[1 1 1 0];
beq=[1];
VLB = [0; 0; 0; -inf];
VUB = [ ];
x0 = [0.2; 0.5; 0.3; 1];
OPT = optimset('LargeScale', 'off');
[x, fval] = fmincon('networkf', x0, [ ], [ ], Aeq, beq, VLB, VUB, 'networknonlcon3', OPT)
```

As it mentioned before, the opposite number of consistency index in the function “network” is taken. The corresponding objective function file “networkf.m” is as follows:

```
function f = networkf(x);
f = -x(4);
```

Since the triangular fuzzy comparison matrix is symmetric, only three constraints above the diagonal need to be considered, that is (1/6, 1/5, 1/4), (1/4, 1/3, 1/2) and (2, 3, 4). The “networknonlcon3.m” file for this matrix is as follows:

```
function [c,ceq] = networknonlcon3(x);
c = [
(1/5-1/6)*x(4)*x(2)-x(1)+(1/6)*x(2);
(1/4-1/5)*x(4)*x(2)+x(1)-(1/4)*x(2);
(1/3-1/4)*x(4)*x(3)-x(1)+(1/4)*x(3);
(1/2-1/3)*x(4)*x(3)+x(1)-(1/2)*x(3);
(3-2)*x(4)*x(3)-x(2)+(2)*x(3);
(4-3)*x(4)*x(3)+x(2)-(4)*x(3);
];
ceq = [];
```

```
>> networkmain
Optimization terminated: first-order optimality measure less
than options.TolFun and maximum constraint violation is less
than options.TolCon.
Active inequalities (to within options.TolCon = 1e-006):
    lower    upper    ineqlin    ineqnonlin
         1
         4
         5

x =

    0.1128
    0.6265
    0.2607
    0.4031

fval =

   -0.4031
```

**Figure 3.** The local weights and consistency index of Example 1

In Fig. 3,  $x$  is the optimal solution and  $fval$  is the optimal value. The first three values of  $x$  are the local weights of the matrix, and the last one is the consistency index.

The final step, is to run “networkmain” in the command window to acquire the local weights. The local weights  $x(1)$ ,  $x(2)$ ,  $x(3)$  are 0.1128, 0.6265, 0.2607 respectively, as shown in Figure 3. The consistency index  $x(4)$  is 0.4031 > 0, which means the fuzzy comparison matrix has a good consistency, and the results are acceptable.

#### 4.1.3. Local weights of a fuzzy pairwise comparison matrix of order 4

**Example 2.** How to solve the local weights of the following fuzzy pairwise comparison matrix of order 4?

$$\begin{pmatrix} (1,1,1) & (3/2,2,5/2) & (1,3/2,2) & (5/2,3,7/2) \\ (2/5,1/2,2/3) & (1,1,1) & (2/3,1,2) & (3/2,2,5/2) \\ (1/2,2/3,1) & (1/2,1,3/2) & (1,1,1) & (2,5/2,3) \\ (2/7,1/3,2/5) & (2/5,1/2,2/3) & (1/3,2/5,1/2) & (1,1,1) \end{pmatrix}$$

The calculation process for a matrix of order 4 is similar to that of a matrix of order 3. A matrix of order 4 has four local weights, named as  $x(1)$ ,  $x(2)$ ,  $x(3)$  and  $x(4)$ , and a consistency index, referred as  $x(5)$ . The following program is the corresponding “networkmain.m” file for this matrix.

```
Aeq=[1 1 1 1 0];
beq=[1];
VLB = [0; 0; 0; 0; -inf];
VUB = [ ];
x0 = [0.1; 0.4; 0.2; 0.3; -1];
OPT = optimset('LargeScale', 'off');
[x, fval] = fmincon('networkf', x0, [ ], [ ], Aeq, beq, VLB, VUB, 'networknonlcon4', OPT)
```

The corresponding objective function file for this matrix is as follows:

```
function f = networkf(x);
f = -x(5);
```

Six constraints for this fuzzy pairwise comparison matrix need to be calculated, that is,  $(3/2, 2, 5/2)$ ,  $(1, 3/2, 2)$ ,  $(5/2, 3, 7/2)$ ,  $(2/3, 1, 2)$ ,  $(3/2, 2, 5/2)$  and  $(2, 5/2, 3)$ . The “networknonlcon4.m” file for this matrix of order 4 is as follows:

```
function [c, ceq] = networknonlcon4(x);
c = [
(2-3/2)*x(5)*x(2) - x(1) + (3/2)*x(2);
(5/2 - 2)*x(5)*x(2) + x(1) - (5/2)*x(2);
(3/2-1)*x(5)*x(3) - x(1) + (1)*x(3);
```

```

(2 - 3/2)*x(5)*x(3) + x(1) - (2)*x(3);
(3-5/2)*x(5)*x(4) - x(1) + (5/2)*x(4);
(7/2-3)*x(5)*x(4) + x(1) - (7/2)*x(4);
(1-2/3)*x(5)*x(3) - x(2) + (2/3)*x(3);
(2-1)*x(5)*x(3) + x(2) - (2)*x(3);
(2-3/2)*x(5)*x(4) - x(2) + (3/2)*x(4);
(5/2-2)*x(5)*x(4) + x(2) - (5/2)*x(4);
(5/2 - 2)*x(5)*x(4) - x(3) + (2)*x(4);
(3 - 5/2)*x(5)*x(4) + x(3) - (3)*x(4);
];
ceq = [ ];

```

Finally, “networkmain” is run in the command panel to obtain the local weights. The optimal solutions are  $x(1)=0.3891$ ,  $x(2)=0.2229$ ,  $x(3)=0.2685$ ,  $x(4)=0.1196$ ,  $x(5)=0.4910$ , as shown in Figure 4. The consistency index  $x(5)$  is  $0.4910 > 0$ , which means the fuzzy comparison matrix has a good consistency, and the results are acceptable.

```

>> networkmain
Optimization terminated: first-order optimality measure less than options.TolFun
and maximum constraint violation is less than options.TolCon.
Active inequalities (to within options.TolCon = 1e-006):
    lower    upper    ineqlin    ineqnonlin
         1
         6
         7
        11

x =

    0.3891
    0.2229
    0.2685
    0.1196
    0.4910

fval =

   -0.4910

```

**Figure 4.** The local weights and consistency index of Example 2

where  $x$  is the optimal solution and  $fval$  is the optimal value. The first four values of  $x$  are the local weights of the matrix, and the last one is the consistency index.



4.1.4. Local weights of a fuzzy pairwise comparison matrix of order  $n$

**Example 3.** How to solve the local weights of the following fuzzy pairwise comparison matrix of order  $n$ ?

$$\begin{pmatrix} (l_{11}, m_{11}, u_{11}) & (l_{12}, m_{12}, u_{12}) & \dots & (l_{1n}, m_{1n}, u_{1n}) \\ (l_{21}, m_{21}, u_{21}) & (l_{22}, m_{22}, u_{22}) & \dots & (l_{2n}, m_{2n}, u_{2n}) \\ \vdots & \vdots & \ddots & \vdots \\ (l_{n1}, m_{n1}, u_{n1}) & (l_{n2}, m_{n2}, u_{n2}) & \dots & (l_{nn}, m_{nn}, u_{nn}) \end{pmatrix}$$

A  $(n \times n)$  matrix has  $(n+1)$  variables, including  $n$  local weights and one consistency index, named as  $x(1), x(2), \dots, x(n+1)$ . Linear equality constraint is as follows:

$$x(1) + x(2) + \dots + x(n) + 0 \cdot x(n+1) = 1$$

Then,  $A_{eq} = [1 \ 1 \ \dots \ 1 \ 0]$ ,  $b_{eq} = [1]$ .

The lower bounds of the local weights are zero, and the lower bound of consistency index is negative infinity. There is no specific upper bound. Then we have

$$VLB = [0; 0; \dots; 0; -inf]; \ VUB = [ \ ]$$

The initial values of the variables can be arbitrary in the range of feasible region. If different initial values lead to different results, it means the nonlinear problem has multiple optimal solutions. Then, we have

$$x_0 = [1; 1; \dots; 1],$$

$$[x, fval] = \text{fmincon}(\text{'networkf'}, x_0, [ \ ], [ \ ], A_{eq}, b_{eq}, VLB, VUB, \text{'networknonlcon'}).$$

The corresponding objective function file `networkf.m` is as follows:

```
function f = networkf(x);
f = -x(n+1);
```

For the non-linear constraints, only those elements above the diagonal need to be considered. That is, these triangular fuzzy numbers  $(l_{ij}, m_{ij}, u_{ij})$  need to be taken into account,  $i < j; i=1, 2, \dots, n; j=2, 3, \dots, n$ .

The “`networknonlcon.m`” file in this case is as follows:

```
function [c,ceq] = networknonlcon(x);
c = [
    (m12-l12)*x(n+1)*x(2)-x(1)+(l12)*x(2);
    (u12-m12)*x(n+1)*x(2)+x(1)-(u12)*x(2);
    (m13-l13)*x(n+1)*x(3)-x(1)+(l13)*x(3);
    (u13-m13)*x(n+1)*x(3)+x(1)-(u13)*x(3);
    ...
    (m1n-l1n)*x(n+1)*x(n)-x(1)+(l1n)*x(n);
```

```

(u1n-m1n)*x(n+1)*x(n)+x(1)-(u1n)*x(n);
(m23-l23)*x(n+1)*x(3)-x(2)+(l23)*x(3);
(u23-m23)*x(n+1)*x(3)+x(2)-(u23)*x(3);
...
(m(n-1)n-l(n-1)n)*x(n+1)*x(n)-x(n-1)+(l(n-1)n)*x(n);
(u(n-1)n-m(n-1)n)*x(n+1)*x(n)+x(n-1)-(u(n-1)n)*x(n);
];
ceq = [ ];

```

Finally, we run “networkmain” in the command panel to obtain the local weights. If the consistency index is positive, the fuzzy comparison matrix has a good consistency and the results are acceptable. Otherwise, we need to modify the fuzzy pairwise comparison matrix until it is satisfied with consistency requirement.

## 4.2. Obtaining limit supermatrix

A limit supermatrix is a weighted supermatrix in a stable state. The weighted supermatrix may be convergent or not. If it is convergent, the limit supermatrix can be achieved. Unfortunately, it is usually not convergent, and then a periodic result is obtained. Under this condition, the limit supermatrix will be achieved only after the periodicity of the supermatrix is determined.

### 4.2.1. The program for acquiring stable limit supermatrix

To obtain a limit supermatrix, four functions named as “fanp\_limitedsupermatrix.m”, “fanp\_multiMatrix.m”, “fanp\_circulantCheck.m” and “fanp\_equal.m” are developed. The first file is the main program by which limit supermatrix can be solved; supermatrix multiplication is implemented by the second file; the third file is used to determine whether a supermatrix is stable or periodic; and the last one is used to test whether a supermatrix after iterations is the same as it was before or not. The second and third files will be called in the main procedure, and the last one will be used in the third program. File “fanp\_limitedsupermatrix.m” is given as follows:

```

function B = fanp_limitedsupermatrix( );
weightedsupermatrix = [0.00000,0.00000,0.00000,0.63400,0.25000,0.40000;
    0.00000,0.00000,0.00000,0.19200,0.25000,0.20000;
    0.00000,0.00000,0.00000,0.17400,0.50000,0.40000;
    0.63700,0.58200,0.13600,0.00000,0.00000,0.00000;
    0.10500,0.10900,0.65400,0.00000,0.00000,0.00000;
    0.25800,0.30900,0.21000,0.00000,0.00000,0.00000];
newMatrix = fanp_multiMatrix(weightedsupermatrix,weightedsupermatrix);
matrixRecord = weightedsupermatrix;
times = 1;
matrixRecord(:, :, 2) = newMatrix;
circulantCheckResult = fanp_circulantCheck(matrixRecord, newMatrix)
while circulantCheckResult(1) == 0

```

```

times = times + 1;
    disp(times)
    newMatrix = fanp_multiMatrix(newMatrix,weightedSupermatrix);
    matrixRecord(:, :, times+1) = newMatrix;
    circulantCheckResult = fanp_circulantCheck(matrixRecord, newMatrix);
end
disp('total multied :')
disp(times)
disp('times')

```

where the “weighted supermatrix” can be arbitrary, which is derived from an unweighted supermatrix. It needs to use a semicolon to separate each row of the supermatrix. “newMatrix” is the new matrix after an iteration. The variable “time” is to keep count of iterations. Variable “matrixRecord” is a three-dimensional variable used to record the output of each iteration.

After each iteration, “matrixRecord” is called to check whether the “newMatrix” is stable or periodic. Whenever the “newMatrix” reaches a stable or periodic state, the program terminates. Otherwise, the program will keep on iterating. Finally, the total number of iterations will be displayed.

The program for a matrix or supermatrix multiplication is as follows:

```

function array3 = fanp_multiMatrix(array1, array2)
    n = size(array1);
    for i = 1: n
        for j = 1: n
            sum = 0;
            for m = 1: n
                sum = sum + array1(i, m)*array2(m, j);
            end
            array3(i, j) = sum;
        end
    end
end

```

where the parameters “array1” and “array2” are the results of current iteration, and the return value “array3” records the result of a new iteration.

The following function “fanp\_circulantCheck” is for determining whether the new supermatrix is stable or periodic. Function “fanp\_equal” will be called in this procedure.

```

function [circulantFlag,rLength,limitedMatrix,cycles] = fanp_circulantCheck(matrixRecord,
newMatrix);
circulantFlag = 0;
limitedMatrix = [ ];
    cycles = 1;
a = size(matrixRecord);

```

```

rLength = a(3);
for i = rLength-1 : -1 : 1
    if fanp_equal(matrixRecord(:, :, i),newMatrix) == 1
        disp('cycle started...')
        circulantFlag = 1;
        limitedMatrix = matrixRecord(:, :, i);
        for j = i + 1 : rLength - 1
            limitedMatrix = limitedMatrix + matrixRecord(:, :, j);
            cycles = cycles + 1;
        end
        limitedMatrix = limitedMatrix / cycles;
        disp('stable matrix : ')
        limitedMatrix
        disp('cycle : ')
        cycles
        disp('cycle start times : ')
        i
        return
    end
end
end

```

In the program above, “matrixRecord” and “newMatrix” are the input parameters, and “circulantFlag” is the output and the sign of a cycle. If the supermatrix has a cycle, “1” is returned. The limit supermatrix, cycles and iterative times of the cycle starting will be displayed.

Comparisons will be made between the iterative output and the existing results one by one. If any two of them are equal, then the cycle of the supermatrix occurs. In this case, the final limit supermatrix is acquired by dividing the summation of all the supermatrices in the cycle by the value of period. For a supermatrix without a cycle, we can assume that its period is one. Therefore, whatever the results of the comparison, we can use the procedure above to obtain the ultimate limit supermatrix.

The following function “fanp\_equal” is used to test whether the new supermatrix is the same as the former one or not.

```

function equalFlag = fanp_equal(array1,array2)
    n = size(array1);
    for i=1: n
        for j=1: n
            if array1(i, j) ~= array2(i, j)
                disp('the two matrix are not equal')
                equalFlag = 0;
                return
            end
        end
    end
end

```

```

end
disp('the two matrix are equal')
equalFlag = 1;
return

```

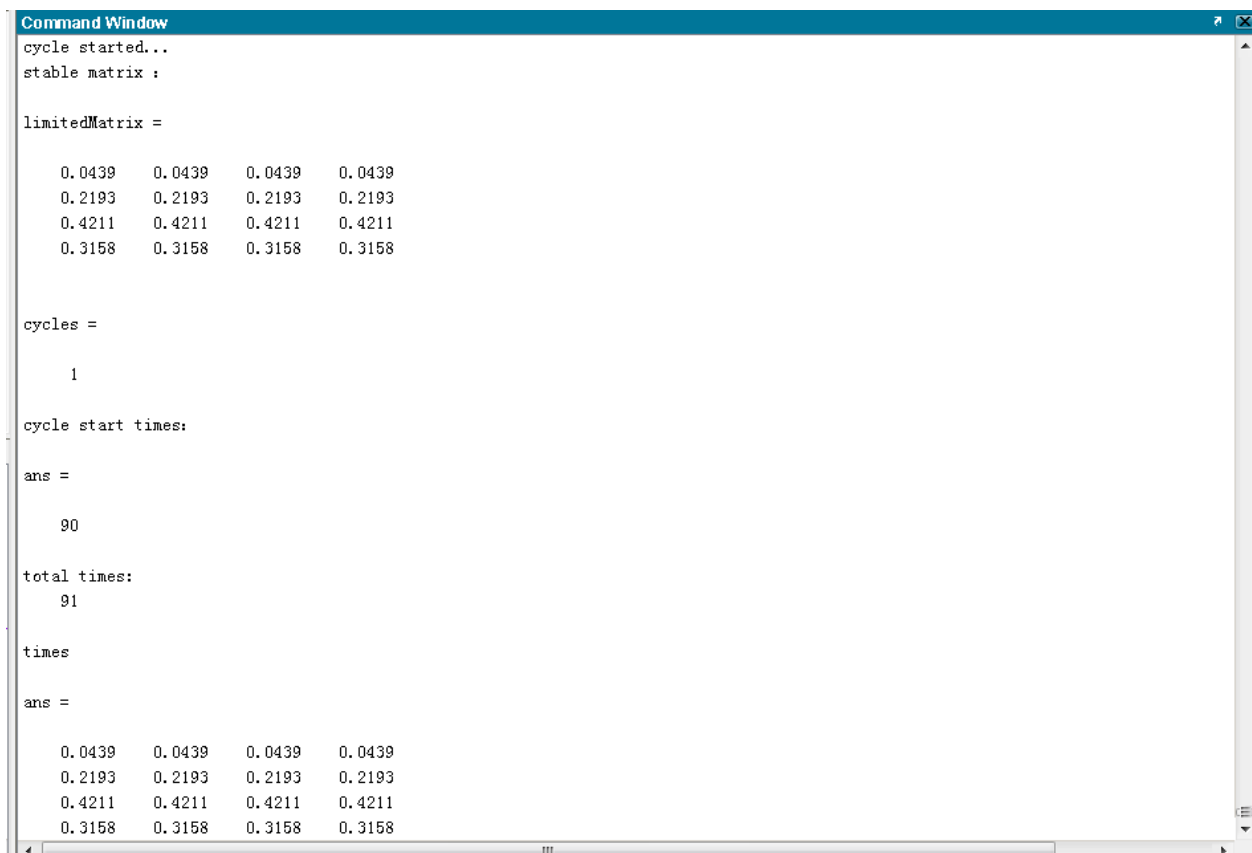
where elements of “array1” and “array2” will be compared one by one. If they are equal, return 1; otherwise, return 0.

#### 4.2.2. Acquiring the limit supermatrix from a supermatrix without a cycle

**Example 4.** Acquiring the limit supermatrix from the following weighted supermatrix.

$$\begin{pmatrix} 0.0000 & 0.2000 & 0.0000 & 0.0000 \\ 0.2000 & 0.0000 & 0.5000 & 0.0000 \\ 0.4000 & 0.4000 & 0.0000 & 1.0000 \\ 0.4000 & 0.4000 & 0.0000 & 0.0000 \end{pmatrix}$$

As mentioned before, the weighted supermatrix can be specified in the function “fanp\_limitedsupermatrix”. The result obtained by running it in the command window, as shown in Fig. 5.



```

Command Window
cycle started...
stable matrix :

limitedMatrix =

    0.0439    0.0439    0.0439    0.0439
    0.2193    0.2193    0.2193    0.2193
    0.4211    0.4211    0.4211    0.4211
    0.3158    0.3158    0.3158    0.3158

cycles =

     1

cycle start times:

ans =

     90

total times:

     91

times

ans =

    0.0439    0.0439    0.0439    0.0439
    0.2193    0.2193    0.2193    0.2193
    0.4211    0.4211    0.4211    0.4211
    0.3158    0.3158    0.3158    0.3158

```

**Figure 5.** The operation result of Example 4

According to the result, the limit supermatrix is

$$\begin{pmatrix} 0.0439 & 0.0439 & 0.0439 & 0.0439 \\ 0.2193 & 0.2193 & 0.2193 & 0.2193 \\ 0.4211 & 0.4211 & 0.4211 & 0.4211 \\ 0.3158 & 0.3158 & 0.3158 & 0.3158 \end{pmatrix}$$

where “cycles” is 1 means the limit supermatrix is not periodic, and “cycle start times” is 90 means the cycle appears at the ninetieth iteration. The total number of iterations is 91. The limit supermatrix is obtained at the end of ninetieth iteration though the program was implemented one more time. The local weights are (0.0439, 0.2193, 0.4211, 0.3158) for this supermatrix.

### 4.2.3. Acquiring the limit supermatrix from a supermatrix with a cycle

**Example 5.** Acquiring the limit supermatrix from the following supermatrix

0.000	0.250	0.266	0.086	0.269	0.100	0.250	0.269	0.000	0.000	0.000	0.000	0.133	0.133	0.083
0.083	0.000	0.067	0.268	0.085	0.200	0.125	0.085	0.000	0.000	0.000	0.000	0.133	0.133	0.167
0.250	0.083	0.000	0.146	0.146	0.200	0.125	0.146	0.000	0.000	0.000	0.000	0.067	0.067	0.083
0.123	0.106	0.104	0.000	0.263	0.132	0.230	0.206	0.290	0.315	0.298	0.250	0.000	0.000	0.000
0.052	0.039	0.033	0.233	0.000	0.084	0.128	0.115	0.032	0.035	0.033	0.028	0.000	0.000	0.000
0.052	0.058	0.104	0.087	0.093	0.000	0.071	0.115	0.065	0.057	0.081	0.120	0.000	0.000	0.000
0.075	0.106	0.059	0.127	0.093	0.230	0.000	0.064	0.081	0.057	0.055	0.074	0.000	0.000	0.000
0.032	0.024	0.033	0.053	0.051	0.054	0.071	0.000	0.032	0.035	0.033	0.028	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.288	0.116	0.086	0.043	0.088	0.058
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.286	0.000	0.321	0.268	0.150	0.154	0.156
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.143	0.157	0.000	0.146	0.097	0.056	0.084
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.071	0.056	0.063	0.000	0.043	0.036	0.036
0.177	0.102	0.102	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.083	0.111
0.100	0.056	0.176	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.111	0.000	0.222
0.056	0.176	0.056	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.223	0.250	0.000

```

Command Window
cycle started...
stable matrix :
limitedMatrix =

0.1372 0.1372 0.1372 0.1372 0.1372 0.1372 0.1372 0.1372 0.1372 0.1372 0.1372 0.1372 0.1372 0.1372 0.1372
0.1064 0.1064 0.1064 0.1064 0.1064 0.1064 0.1064 0.1064 0.1064 0.1064 0.1064 0.1064 0.1064 0.1064 0.1064
0.1091 0.1091 0.1091 0.1091 0.1091 0.1091 0.1091 0.1091 0.1091 0.1091 0.1091 0.1091 0.1091 0.1091 0.1091
0.1231 0.1231 0.1231 0.1231 0.1231 0.1231 0.1231 0.1231 0.1231 0.1231 0.1231 0.1231 0.1231 0.1231 0.1231
0.0655 0.0655 0.0655 0.0655 0.0655 0.0655 0.0655 0.0655 0.0655 0.0655 0.0655 0.0655 0.0655 0.0655 0.0655
0.0588 0.0588 0.0588 0.0588 0.0588 0.0588 0.0588 0.0588 0.0588 0.0588 0.0588 0.0588 0.0588 0.0588 0.0588
0.0729 0.0729 0.0729 0.0729 0.0729 0.0729 0.0729 0.0729 0.0729 0.0729 0.0729 0.0729 0.0729 0.0729 0.0729
0.0327 0.0327 0.0327 0.0327 0.0327 0.0327 0.0327 0.0327 0.0327 0.0327 0.0327 0.0327 0.0327 0.0327 0.0327
0.0292 0.0292 0.0292 0.0292 0.0292 0.0292 0.0292 0.0292 0.0292 0.0292 0.0292 0.0292 0.0292 0.0292 0.0292
0.0478 0.0478 0.0478 0.0478 0.0478 0.0478 0.0478 0.0478 0.0478 0.0478 0.0478 0.0478 0.0478 0.0478 0.0478
0.0276 0.0276 0.0276 0.0276 0.0276 0.0276 0.0276 0.0276 0.0276 0.0276 0.0276 0.0276 0.0276 0.0276 0.0276
0.0132 0.0132 0.0132 0.0132 0.0132 0.0132 0.0132 0.0132 0.0132 0.0132 0.0132 0.0132 0.0132 0.0132 0.0132
0.0578 0.0578 0.0578 0.0578 0.0578 0.0578 0.0578 0.0578 0.0578 0.0578 0.0578 0.0578 0.0578 0.0578 0.0578
0.0586 0.0586 0.0586 0.0586 0.0586 0.0586 0.0586 0.0586 0.0586 0.0586 0.0586 0.0586 0.0586 0.0586 0.0586
0.0601 0.0601 0.0601 0.0601 0.0601 0.0601 0.0601 0.0601 0.0601 0.0601 0.0601 0.0601 0.0601 0.0601 0.0601

cycles =

2

cycle start times:

ans =

72
    
```

Figure 6. The operation result of Example 5

Specify the weighted supermatrix in the function “fanp\_limitedsupermatrix” and run the program in the command window, the result is shown in Fig. 6. Where “cycles” is 2 means the limit supermatrix is periodic, and the period is 2. “cycle start times” is 72 means the cycle appears since the 72nd iteration, and the limit supermatrix is obtained at the end of the 73rd iteration. According to the FPP method, the local weights are (0.1372, 0.1064, 0.1091, 0.1231, 0.0655, 0.0588, 0.0729, 0.0327, 0.0292, 0.0478, 0.0276, 0.0132, 0.0578, 0.0586, 0.0601). This example is actually the weighted supermatrix of the following case in section 5, and the result is the limit supermatrix of the case.

## 5. Case study

Supposing that a company has the opportunity to select an optimal project from a number of alternatives. Through pre-test, three projects, named as  $D_1$ ,  $D_2$  and  $D_3$ , need further evaluation. A cross-functional project team consists of various departments working to select the best project. Firstly the selection criteria are identified. Then according to the FANP method, the optimal alternative is derived. The decision-making process of choosing an optimal project based on FANP is as follows:

**Step 1.** Model construction and problem structuring. Taking the interaction among dimensions and attribute-enablers into account, a four-level evaluation index system is proposed, as shown in Fig. 2.

**Step 2.** Pairwise comparison matrices among dimensions/attributes are formed by the decision committee using the scales given in Table 1, and the scores of the three projects are determined as well. For instance, Table 2. is the pairwise comparison matrix for the profitability ( $S_1$ ), risk ( $S_2$ ), owners ( $S_3$ ) and Bidding competition ( $S_4$ ) at the dimension of choosing an optimal project. All the fuzzy comparison matrices are produced in the same way.

optimal project	$S_1$	$S_2$	$S_3$	$S_4$
$S_1$	EI	$1/IM_2$	$1/MI$	$1/IM_1$
$S_2$	$IM_2$	EI	$IM_1$	$IM_1$
$S_3$	$MI$	$1/IM_1$	EI	$1/IM_1$
$S_4$	$IM_1$	$1/IM_1$	$IM_1$	EI

**Table 2.** The comparison matrix using linguistic scales at the dimension of optimal project

Expert opinions will be converted into the corresponding triangular fuzzy numbers, as shown in Table 3.

optimal project	$S_1$	$S_2$	$S_3$	$S_4$	Local weights
$S_1$	(1, 1, 1)	(1/5, 1/4, 1/3)	(1/4, 1/3, 1/2)	(1/3, 1/2, 1)	0.0989
$S_2$	(3, 4, 5)	(1, 1, 1)	(1, 2, 3)	(1, 2, 3)	0.4240
$S_3$	(2, 3, 4)	(1/3, 1/2, 1)	(1, 1, 1)	(1/3, 1/2, 1)	0.2544
$S_4$	(1, 2, 3)	(1/3, 1/2, 1)	(1, 2, 3)	(1, 1, 1)	0.2226
$\lambda=0.6667$					

**Table 3.** The comparison matrix using TFNs at the dimension of optimal project

**Step 3.** Local weights of the factors and sub-factors which take part in the second and third levels of the ANP model, provided in Fig. 2, are calculated by FPP method. For example, according to equation (8), the local weights of Table 3 can be achieved by solving the following non-linear programming.

$$\begin{aligned}
 & \text{Max } \lambda \\
 & (1/20)\lambda w_2 - w_1 + (1/5)w_2 \leq 0; \\
 & (1/12)\lambda w_2 + w_1 - (1/3)w_2 \leq 0; \\
 & (1/12)\lambda w_3 - w_1 + (1/4)w_3 \leq 0; \\
 & (1/6)\lambda w_3 + w_1 - (1/2)w_3 \leq 0; \\
 & (1/6)\lambda w_4 - w_1 + (1/3)w_4 \leq 0; \\
 & (1/2)\lambda w_4 + w_1 - w_4 \leq 0; \\
 & \lambda w_3 - w_2 + w_3 \leq 0; \\
 & \lambda w_3 + w_2 - 3w_3 \leq 0; \\
 & \lambda w_4 - w_2 + w_4 \leq 0; \\
 & \lambda w_4 + w_2 - 3w_4 \leq 0; \\
 & (1/6)\lambda w_4 - w_3 + (1/3)w_4 \leq 0; \\
 & (1/2)\lambda w_4 + w_3 - w_4 \leq 0; \\
 & w_1 + w_2 + w_3 + w_4 = 1; \\
 & w_1, w_2, w_3, w_4 \geq 0.
 \end{aligned}$$

	S <sub>11</sub>	S <sub>12</sub>	S <sub>13</sub>	S <sub>21</sub>	S <sub>22</sub>	S <sub>23</sub>	S <sub>24</sub>	S <sub>25</sub>	S <sub>31</sub>	S <sub>32</sub>	S <sub>33</sub>	S <sub>34</sub>	S <sub>41</sub>	S <sub>42</sub>	S <sub>43</sub>
S <sub>11</sub>	0.000	0.750	0.800	0.171	0.538	0.200	0.500	0.538	0.000	0.000	0.000	0.000	0.400	0.400	0.250
S <sub>12</sub>	0.250	0.000	0.200	0.536	0.170	0.400	0.250	0.170	0.000	0.000	0.000	0.000	0.400	0.400	0.500
S <sub>13</sub>	0.750	0.250	0.000	0.293	0.293	0.400	0.250	0.293	0.000	0.000	0.000	0.000	0.200	0.200	0.250
S <sub>21</sub>	0.368	0.318	0.313	0.000	0.526	0.263	0.458	0.412	0.581	0.631	0.595	0.500	0.000	0.000	0.000
S <sub>22</sub>	0.155	0.118	0.099	0.467	0.000	0.169	0.256	0.230	0.065	0.070	0.066	0.056	0.000	0.000	0.000
S <sub>23</sub>	0.155	0.173	0.313	0.174	0.186	0.000	0.143	0.230	0.129	0.115	0.162	0.241	0.000	0.000	0.000
S <sub>24</sub>	0.226	0.318	0.176	0.253	0.186	0.460	0.000	0.128	0.161	0.115	0.110	0.148	0.000	0.000	0.000
S <sub>25</sub>	0.095	0.073	0.099	0.107	0.102	0.108	0.143	0.000	0.065	0.070	0.066	0.056	0.000	0.000	0.000
S <sub>31</sub>	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.575	0.231	0.171	0.130	0.263	0.174
S <sub>32</sub>	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.571	0.000	0.644	0.536	0.450	0.460	0.467
S <sub>33</sub>	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.286	0.314	0.000	0.293	0.290	0.169	0.253
S <sub>34</sub>	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.143	0.111	0.125	0.000	0.130	0.108	0.107
S <sub>41</sub>	0.535	0.307	0.306	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.250	0.333
S <sub>42</sub>	0.299	0.168	0.527	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.333	0.000	0.667
S <sub>43</sub>	0.167	0.525	0.167	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.667	0.750	0.000

**Table 4.** The unweighted supermatrix

As mentioned before, the non-linear programming can be solved by Matlab. The optimal solutions are  $w_1=0.0989$ ,  $w_2=0.4240$ ,  $w_3=0.2544$ ,  $w_4=0.2226$ , and  $\lambda=0.6667$ , which shows the



experts' opinions are of good consistency, and the comparison result is acceptable, as shown in Table 3. All the local weights are acquired in the same manner.

**Step 4.** According to the interdependencies among dimensions and attribute-enablers, an unweighted supermatrix is built, as shown in Table 4.

**Step 5.** Randomize the unweighted supermatrix to derive the weighted supermatrix.

**Step 6.** Multiply the weighted supermatrix by itself until the values of each row converge to the same value for every column of the supermatrix. Then we choose any column from the stable limit supermatrix as the local weights of interdependency indicators, as shown in Table 5. It can be solved by Matlab, and the process of calculation is the same as Example 5 in the former section.

	S <sub>11</sub>	S <sub>12</sub>	S <sub>13</sub>	S <sub>21</sub>	S <sub>22</sub>	S <sub>23</sub>	S <sub>24</sub>	S <sub>25</sub>	S <sub>31</sub>	S <sub>32</sub>	S <sub>33</sub>	S <sub>34</sub>	S <sub>41</sub>	S <sub>42</sub>	S <sub>43</sub>
S <sub>11</sub>	0.137	0.137	0.137	0.137	0.137	0.137	0.137	0.137	0.137	0.137	0.137	0.137	0.137	0.137	0.137
S <sub>12</sub>	0.106	0.106	0.106	0.106	0.106	0.106	0.106	0.106	0.106	0.106	0.106	0.106	0.106	0.106	0.106
S <sub>13</sub>	0.109	0.109	0.109	0.109	0.109	0.109	0.109	0.109	0.109	0.109	0.109	0.109	0.109	0.109	0.109
S <sub>21</sub>	0.123	0.123	0.123	0.123	0.123	0.123	0.123	0.123	0.123	0.123	0.123	0.123	0.123	0.123	0.123
S <sub>22</sub>	0.066	0.066	0.066	0.066	0.066	0.066	0.066	0.066	0.066	0.066	0.066	0.066	0.066	0.066	0.066
S <sub>23</sub>	0.059	0.059	0.059	0.059	0.059	0.059	0.059	0.059	0.059	0.059	0.059	0.059	0.059	0.059	0.059
S <sub>24</sub>	0.073	0.073	0.073	0.073	0.073	0.073	0.073	0.073	0.073	0.073	0.073	0.073	0.073	0.073	0.073
S <sub>25</sub>	0.033	0.033	0.033	0.033	0.033	0.033	0.033	0.033	0.033	0.033	0.033	0.033	0.033	0.033	0.033
S <sub>31</sub>	0.029	0.029	0.029	0.029	0.029	0.029	0.029	0.029	0.029	0.029	0.029	0.029	0.029	0.029	0.029
S <sub>32</sub>	0.048	0.048	0.048	0.048	0.048	0.048	0.048	0.048	0.048	0.048	0.048	0.048	0.048	0.048	0.048
S <sub>33</sub>	0.028	0.028	0.028	0.028	0.028	0.028	0.028	0.028	0.028	0.028	0.028	0.028	0.028	0.028	0.028
S <sub>34</sub>	0.013	0.013	0.013	0.013	0.013	0.013	0.013	0.013	0.013	0.013	0.013	0.013	0.013	0.013	0.013
S <sub>41</sub>	0.058	0.058	0.058	0.058	0.058	0.058	0.058	0.058	0.058	0.058	0.058	0.058	0.058	0.058	0.058
S <sub>42</sub>	0.059	0.059	0.059	0.059	0.059	0.059	0.059	0.059	0.059	0.059	0.059	0.059	0.059	0.059	0.059
S <sub>43</sub>	0.060	0.060	0.060	0.060	0.060	0.060	0.060	0.060	0.060	0.060	0.060	0.060	0.060	0.060	0.060

**Table 5.** The limit supermatrix

**Step 7.** Calculate the comprehensive weights of each index, as shown in Table 6.

**Step 8.** According to equation (11), the scores of each alternative can be calculated,  $D_1=0.31$ ,  $D_2=0.33$ ,  $D_3=0.36$ , as shown in Table 6. Therefore, we can choose project  $D_3$  as the best one.

## 6. Conclusions

Taking the interaction and feedback relationships between criteria and/or indicators into account, an evaluation index system for selecting a construction project is proposed. With the uncertain and inaccurate information during the evaluation process being considered, an evaluation and selection model based on fuzzy analytic network process method is presented. The weights of the indices, including the weights of the criterion level indicators, the weights of independent sub-indices and the weights of dependent sub-indices are determined by the fuzzy preference programming method. Meanwhile, an unweighted supermatrix based on its network structure is built for interactional indicators, and the convergent limit supermatrix is calculated after randomizing the unweighted supermatrix. Accordingly, the comprehensive weight of each index and the final score of each alternative

can be calculated. Then we can choose the optimal alternative. A numerical example is given by the proposed method as well.

Two key steps in the process of decision-making with FANP are solved by Matlab. One is acquiring local weights of the fuzzy pairwise comparison matrices; the other is obtaining the limit supermatrix. Matlab is selected for its excellent performance on data processing and matrix operation. Compared with the existing research results, the proposed method fully takes into consideration the interaction and feedback relationships between the dimensions and/or attributes, and it uses triangular fuzzy numbers to represent the preference opinions of experts. It helps to make a more accurate and scientific decision.

index	$P_j$	$A_{ij}^I$	$A_{ij}^D$	$w$	$w'$	$S_{1ij}$	$S_{2ij}$	$S_{3ij}$	$d_1$	$d_2$	$d_3$
S <sub>11</sub>	0.099	0.538	0.137	0.007	0.104	0.381	0.333	0.286	0.039	0.035	0.030
S <sub>12</sub>	0.099	0.170	0.106	0.002	0.025	0.300	0.300	0.400	0.008	0.008	0.010
S <sub>13</sub>	0.099	0.293	0.109	0.003	0.045	0.263	0.368	0.368	0.012	0.017	0.017
S <sub>21</sub>	0.424	0.361	0.123	0.019	0.268	0.286	0.333	0.381	0.076	0.089	0.102
S <sub>22</sub>	0.424	0.243	0.066	0.007	0.096	0.267	0.400	0.333	0.026	0.038	0.032
S <sub>23</sub>	0.424	0.147	0.059	0.004	0.052	0.304	0.304	0.391	0.016	0.016	0.020
S <sub>24</sub>	0.424	0.147	0.073	0.005	0.065	0.214	0.357	0.429	0.014	0.023	0.028
S <sub>25</sub>	0.424	0.102	0.033	0.001	0.020	0.375	0.250	0.375	0.008	0.005	0.008
S <sub>31</sub>	0.254	0.228	0.029	0.002	0.024	0.273	0.318	0.409	0.007	0.008	0.010
S <sub>32</sub>	0.254	0.571	0.048	0.007	0.099	0.364	0.318	0.318	0.036	0.031	0.031
S <sub>33</sub>	0.254	0.124	0.028	0.001	0.012	0.286	0.333	0.381	0.004	0.004	0.005
S <sub>34</sub>	0.254	0.077	0.013	0.000	0.004	0.333	0.286	0.381	0.001	0.001	0.001
S <sub>41</sub>	0.223	0.170	0.058	0.002	0.031	0.400	0.250	0.350	0.012	0.008	0.011
S <sub>42</sub>	0.223	0.300	0.059	0.004	0.056	0.421	0.263	0.316	0.023	0.015	0.018
S <sub>43</sub>	0.223	0.529	0.060	0.007	0.101	0.286	0.333	0.381	0.029	0.034	0.038
The score of alternative $D_k$									0.31	0.33	0.36

**Table 6.** The comprehensive weights and the ranking of the alternatives

## Author details

Xiaoguang Zhou

*Dongling School of Economics and Management, University of Science and Technology Beijing, Beijing China*

## Acknowledgement

The author is very grateful to the editor, Dr. Vasilios Katsikis for his valuable suggestions, and Robert Ulbrich who gives constructive comments and suggestions on English grammar,

which have been very helpful in improving the book. This work was supported by “the Fundamental Research Funds for Chinese Central Universities (No. FRF-BR-11-009A)”.

## 7. References

- Amiri, Morteza Pakdin. (2010). Project selection for oil-fields development by using the AHP and fuzzy TOPSIS methods. *Expert Systems with Applications*, vol. 37, pp. 6218-6224
- Aragonés-Beltrán, P.; Chaparro-González, F.; Pastor-Ferrando, J. P. & Rodríguez-Pozo, F. (2010). An ANP-based approach for the selection of photovoltaic solar power plant investment projects. *Renewable and Sustainable Energy Reviews*, vol. 14, pp. 249-264
- Arunraj, N. S. & Maiti, J. (2010). Risk-based maintenance policy selection using AHP and goal programming. *Safety Science*, vol. 48, pp. 238-247
- Ayağ, Z. & Özdemir, R. G. (2009). A hybrid approach to concept selection through fuzzy analytic network process. *Computers & Industrial Engineering*, vol. 56, pp. 368-379
- Bhattacharyya, Rupak.; Kumar, Pankaj. & Kar, Samarjit. (2011). Fuzzy R&D portfolio selection of interdependent projects. *Computers and Mathematics with Applications*, vol. 62, pp. 3857-3870
- Boran, Semra. & Goztepe, Kerim. (2010). Development of a fuzzy decision support system for commodity acquisition using fuzzy analytic network process. *Expert Systems with Applications*, vol. 37, pp. 1939-1945
- Buckley, J. J. (1985). Fuzzy hierarchical analysis. *Fuzzy Sets and Systems*, vol. 17, pp. 233-247
- Chang, D. Y. (1996). Applications of the extent analysis method on fuzzy AHP. *European Journal of Operational Research*, vol. 95, pp. 649-655
- Chang, P. T. & Lee, J. H. (2012). A fuzzy DEA and knapsack formulation integrated model for project selection. *Computers & Operations Research*, vol. 39, pp. 112-125
- Cheng, E. W. L. & Li, H. (2005). Analytic Network Process Applied to Project Selection. *Journal of Construction Engineering and Management*, vol. 131 (4), pp. 459-466.
- Csutora, R. & Buckley, J. J. (2001). Fuzzy hierarchical analysis: The Lamda-Max method. *Fuzzy Sets and Systems*, vol. 120, pp. 181-195
- Dağdeviren, Metin. & Yüksel, İhsan. (2010). A fuzzy analytic network process (ANP) model for measurement of the sectoral competitiveness level (SCL). *Expert Systems with Applications*, vol. 37, pp. 1005-1014
- Dey, Prasanta Kumar. (2006). Integrated project evaluation and selection using multiple-attribute decision-making technique. *Int. J. Production Economics*, vol. 103, pp. 90-103
- Ebrahimnejad, S.; Mousavi, S. M.; Tavakkoli-Moghaddam, R.; Hashemi, H. & Vahdani, B. (2011). A novel two-phase group decisionmaking approach for construction project selection in a fuzzy environment. *Applied Mathematical Modelling*. doi: 10. 1016/j. apm. 2011. 11. 050
- Fang, Yong.; Chen, Lihua. & Fukushima, Masao. (2008). A mixed R&D projects and securities portfolio selection model. *European Journal of Operational Research*, vol. 185, pp. 700-715

- Gabriel, Steven A.; Kumar, Satheesh.; Ordóñez, Javier. & Nasserian, Amirali. (2006). A multiobjective optimization model for project selection with probabilistic considerations. *Socio-Economic Planning Sciences*, vol. 40, pp. 297-313
- Ghorbani, S. & Rabbani, M. (2009). A new multi-objective algorithm for a project selection problem. *Advances in Engineering Software*, vol. 40, pp. 9-14
- Gutjahr, Walter J.; Katzensteiner, Stefan.; Reiter, Peter.; Stummer, Christian. & Denk, Michaela. (2010). Multi-objective decision analysis for competence-oriented project portfolio selection. *European Journal of Operational Research*, vol. 205, pp. 670-679
- Huang, Chi-Cheng.; Chu, Pin-Yu. & Chiang, Yu-Hsiu. (2008). A fuzzy AHP application in government-sponsored R&D project selection. *Omega*, vol. 36, pp. 1038-1052
- Huang, Ivy B.; Keisler, Jeffrey. & Linkov Igor. (2011). Multi-criteria decision analysis in environmental sciences- Ten years of applications and trends. *Science of the Total Environment*, vol. 409, pp. 3578-3594
- Huo, Liang-an.; Lan, Jibin. & Wang, Zhongxing. (2011). New parametric prioritization methods for an analytical hierarchy process based on a pairwise comparison matrix. *Mathematical and Computer Modelling*, vol. 54, pp. 2736-2749
- Ju, Yanbing.; Wang, Aihua. & Liu, Xiaoyue. (2012). Evaluating emergency response capacity by fuzzy AHP and 2-tuple fuzzy linguistic approach. *Expert Systems with Applications*, vol. 39, pp. 6972-6981
- Jung, U. & Seo, D. W. (2010). An ANP approach for R&D project evaluation based on interdependencies between research objectives and evaluation criteria. *Decision Support Systems*, vol. 49, pp. 335-342
- Kahraman, Cengiz.; Ertay, Tijen. & Büyüközkan, Gülçin. (2006). A fuzzy optimization model for QFD planning process using analytic network approach. *European Journal of Operational Research*, vol. 171, pp. 390-411
- Kumar, D. P. (2006). Integrated project evaluation and selection using multiple-attribute decision-making technique. *International Journal of Production Economics*, vol. 103, pp. 90-103
- Lee, Hakyeon.; Kim, Chulhyun.; Cho, Hyunmyung. & Park, Yongtae. (2009). An ANP-based technology network for identification of core technologies: A case of telecommunication technologies. *Expert Systems with Applications*, vol. 36: 894-908.
- Liesiö, Juuso.; Mild, Pekka. & Salo, Ahti. (2007). Preference programming for robust portfolio modeling and project selection. *European Journal of Operational Research*, vol. 181, pp. 1488-1505
- Lin, C. T. & Chen, Y. T. (2004). Bid/no-bid decision-making-a fuzzy linguistic approach. *International Journal of Project Management*, vol. 22, pp. 585-593
- Mikhailov, L. (2003). Deriving priorities from fuzzy pairwise comparison judgements. *Fuzzy Sets and Systems*, vol. 134, pp. 365-385
- Mikhailov, L. (2004). Group prioritization in the AHP by fuzzy preference programming method. *Computers & Operations Research*, vol. 31, pp. 293-301
- Mohanty, R. P. (1992). Project selection by a multiple-criteria decision making method: An example from a developing country. *International Journal of Project Management*, vol. 10, pp. 31-38

- Pires, Ana.; Chang, Ni-Bin. & Martinho, Graça. (2011). An AHP-based fuzzy interval TOPSIS assessment for sustainable expansion of the solid waste management system in Setúbal Peninsula, Portugal. *Resources, Conservation and Recycling*, vol. 56, pp. 7-21
- Promentilla, Michael Angelo B.; Furuichi, T. ; Ishii, K. & Tanikawa, N. (2008). A fuzzy analytic network process for multi-criteria evaluation of contaminated site remedial countermeasures. *Journal of Environmental Management*, vol. 88, pp. 479–495
- Saaty, T. L. (1996). *Decision-making with Dependence and Feedback: The Analytic Network Process*, RWS Publications, Pittsburgh, PA
- Shakhsi-Niaei, M.; Torabi, S. A. & Iranmanesh, S. H. (2011). A comprehensive framework for project selection problem under uncertainty and real-world constraints. *Computers & Industrial Engineering*, vol. 61, pp. 226-237
- Smith-Perera, Aida.; García-Melón, Mónica.; Poveda-Bautista, Rocío. & Pastor-Ferrando, Juan-Pascual. (2010). A Project Strategic Index proposal for portfolio selection in electrical company based on the Analytic Network Process. *Renewable and Sustainable Energy Reviews*, vol. 14, pp. 1569-1579
- Srdjevic, Bojan. (2005). Combining different prioritization methods in the-analytic hierarchy process synthesis. *Computers & Operations Research*, vol. 32, pp. 1897-1919
- Van Laarhoven, P. J. M. & Pedrycz, W. (1983). A fuzzy extension of Saaty's priority theory. *Fuzzy Sets and Systems*, vol. 11, pp. 229-241
- Wang, J. & Hwang, W. L. (2007). A fuzzy set approach for R&D portfolio selection using a real option valuation model. *Omega*, vol. 35, pp. 247-57
- Wang, J.; Xu, Y. J. & Li, Z. (2009). Research on project selection system of pre-evaluation of engineering design project bidding. *International Journal of Project Management*, vol. 27, pp. 584-599
- Wang, Y. M.; Elhag, T. M. S. & Hua, Z. S. (2006). A modified fuzzy logarithmic least squares method for fuzzy analytic hierarchy process. *Fuzzy Sets and Systems*, vol. 157, pp. 3055-3071
- Xu, R. (2000). Fuzzy least-squares priority method in the analytic hierarchy process. *Fuzzy Sets and Systems*, vol. 112, pp. 359-404
- Yu, Jing-Rung. & Cheng, Sheu-Ji. (2007). An integrated approach for deriving priorities in analytic network process. *European Journal of Operational Research*, vol. 180, pp. 1427-1432
- Zadeh, L. A. (1965). Fuzzy sets. *Information and Control*, vol. 8, pp. 338-353