A MATRIX METHOD FOR THE FUZZY ANALYTIC NETWORK PROCESS

Jih-Jeng Huang
Department of Business and Entrepreneurial Management, Kainan University
No.1 Kainan Rd, Shinshing Tsuen, Luchu Shiang, Taoyuan, Taiwan.

Received 17 May 2007
Revised 7 November 2007
Accepted 13 March 2008

In this paper, the fuzzy analytic network process (FANP) is proposed. For achieving this purpose, two problems are highlighted and overcome in this paper. First, the postulate of the reciprocal matrix should be released, because this property is not satisfied in the fuzzy comparison matrix. Second, the convergent problem for raising the fuzzy supermatrix to limiting power should be appropriately handled. In this paper, we directly fuzzify Cogger and Yu’s method for obtaining the fuzzy local vectors, because their method releases the postulate of the reciprocal matrix in the analytic hierarchy process (AHP). Then, we derive the particular matrix problem for obtaining the fuzzy global weight vector so that the convergent problem in a fuzzy limiting supermatrix can be overcome.

Keywords: Fuzzy analytic network process (FANP); reciprocal matrix; analytic hierarchy process (AHP); fuzzy matrix.

1. Introduction

In 1970s, the analytic hierarchy process (AHP) was proposed by Saaty to model subjective decision-making processes based on the information of pairwise comparison between criteria in a hierarchical system. From that moment on, it has been widely used in dealing with various kinds of multiple criteria decision making (MCDM) problems. However, the restricted postulate of the hierarchical structure makes the AHP cannot well perform in some realistic problems. Therefore, in 1996, Saaty proposed the analytic network process (ANP) to release the postulate of the hierarchical structure so that dependence and feedback effects among criteria can be considered.

In the traditional AHP/ANP, a decision-maker is first asked to express his/her preferences through the exact relative ratios of weights between each criterion, and then, the weight of each criterion is derived. However, since AHP/ANP is a subjective decision-making tool, the restrictions of incomplete information and subjective uncertainty should be considered. Therefore, it is more natural that a decision-making provides fuzzy judgments, instead of precise comparison, to give the relative ratios of weights between each criterion.

To extend the AHP under fuzzy environments, a considerable number of methods have been proposed. In the fuzzy analytic hierarchy process (FAHP), scholars proposed the fuzzy logarithmic least squares method (LLSM), geometric mean method, Lambda-Max method, extent analysis method, and fuzzy preference programming method to...
Jih-Jeng Huang
deal with the problem of uncertainty. Despite many methods were proposed to provide
more accurate and appropriate fuzzy weights, the critical assumption of the reciprocal
matrix seems to be ignored in developing the FAHP. It should be highlighted that the
property of reciprocity truly deters the extension of the FAHP, because if a triangular
fuzzy number \( \tilde{a}_{ij} \) denotes the ratio of the weight that the criterion \( i \) dominates the
criterion \( j \), the symmetric entry \( \tilde{a}_{ji} \) is not triangular (i.e., \( \tilde{a}_{ji} \neq 1/\tilde{a}_{ij} \)). Therefore, we
cannot develop the FAHP based on the incorrect postulation of the reciprocal matrix.

On the other hand, Mikhailov and Singh\(^1\) proposed the fuzzy analytic network
process (FANP) to extend the ANP fuzzy environments. Their method first derived crisp
local weights from the fuzzy pairwise judgments by using the fuzzy preference
programming (FPP) method\(^9\), and then a crisp weighted supermatrix is formed and raised
to a steady-state process to obtain global weights. In other words, their method can only
derive fuzzy weights in the AHP rather than fuzzy weights in the ANP. In addition,
scholars\(^11\)-\(^14\) proposed another method to deal with the uncertain judgments in the ANP
based on fuzzy arithmetic operations. However, their methods may result in the
convergent and rational problems of fuzzy global weights, due to using the standard
fuzzy arithmetic operations to multiple and divide fuzzy numbers. Clearly, the hardest
problem for proposing the FANP is to deal with the convergent problem of a fuzzy
supermatrix, i.e., to overcome the limiting power of a fuzzy supermatrix\(^15\). A simple
example is used to show the convergent problem in a fuzzy limiting matrix as follows.

Note that for simplicity, a fuzzy number used in this paper is represented as a triangular
form.

Let a fuzzy matrix be given as:
\[
M = \begin{bmatrix}
(0.2, 0.3, 0.4) & (0.5, 0.6, 0.7) \\
(0.6, 0.7, 0.8) & (0.3, 0.4, 0.5)
\end{bmatrix}
\]

Then, the limiting matrix of \( \tilde{M} \) can be calculated using interval arithmetic operations
as:
\[
\lim_{k \to \infty} \tilde{M}^{(k)} = \begin{bmatrix}
(0, 0.4615, \infty) & (0, 0.4615, \infty) \\
(0, 0.5385, \infty) & (0, 0.5385, \infty)
\end{bmatrix},
\]

where \((k)\) denotes the power operation. Obviously, the above solution is not a
satisfactory result.

In this paper, we overcome the previous problems as the following way. First, Cogger
and Yu’s eigenvector method\(^16\) is fuzzified to obtain fuzzy local weight vectors so that
the postulate of the reciprocal matrix in the AHP is released. Then, a matrix method is
derived to avoid the convergent problem of a fuzzy supermatrix for obtaining fuzzy
global weights. Hence, the proposed method can provide appropriate fuzzy local weights
but does not claim the problematic postulate that \( \tilde{a}_{ij} = 1/\tilde{a}_{ji} \) and overcome the
convergent problem in raising a fuzzy supermatrix to limiting power.

The remainder of this paper is organized as follows. In Section 2, a matrix method for
the ANP is proposed. The way to develop the FANP is given in Section 3. An application
used to demonstrate the proposed method is in Section 4. Discussions are presented in
Section 5 and conclusions are in the last section.
2. A Matrix Method for the Analytic Network Process

The ANP is a popular technique used to model subjective decision-making processes according to the information of the pairwise comparison between criteria. The first step of the ANP is to compare the ratios of weights between criteria with respect to each cluster. In this step, the postulate of a reciprocal matrix is claimed such that if $a_{ij}$ denotes the ratio of the weight that the $i$th criterion dominates the $j$th criterion, $a_{ji} = 1/a_{ij}$ should be satisfied. Next, in order to derive the local weight vectors, Saaty proposed the eigenvector method (EM) to solve the eigenvector problem. However, as mentioned previously, Saaty’s method cannot work under fuzzy environments, because the property of the reciprocal matrix is not hold in a fuzzy matrix. Therefore, in this paper, Cogger and Yu’s method is introduced to show how the eigenvector can be derived when the postulation of reciprocity in a comparison matrix is released as follows.

Let a positive upper triangular comparison matrix

$$A = [a_{ij}]_{n \times n}, \quad a_{ij} = \begin{cases} a_{ij} & \text{if } i \leq j, \\ 0 & \text{otherwise}, \end{cases}$$

(1)

where $a_{ij}$ denotes the strength of the ratio of the weight that the criterion $i$ dominates the criterion $j$.

Let $D$ be the diagonal matrix such that

$$D = [d_{ij}]_{n \times n}, \quad d_{ij} = \begin{cases} n - i + 1 & \text{if } i = j, \\ 0 & \text{otherwise}. \end{cases}$$

(2)

By introducing the weight vector $w' = (w_1, w_2, ..., w_n)$, we can obtain

$$Aw = Dw$$

(3)

and

$$(D - A - I)w = 0,$$

(4)

where

$$d_{ij}' = \begin{cases} 1/n - i + 1 & \text{if } i = j, \\ 0 & \text{otherwise}. \end{cases}$$

(5)

Next, we incorporate the constraint $w'1 = 1$, where $1' = (1, 1, ..., 1)$, to Eq. (4) and rearrange the matrix such that

$$A'w = e,$$

(6)

where

\[
A' = \begin{bmatrix}
1 - n & a_{12} & \cdots & a_{1k-1} & a_{1n} \\
0 & 2 - n & \cdots & a_{2k-1} & a_{2n} \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & \cdots & -1 & a_{n-1n} \\
1 & 1 & \cdots & 1 & 1
\end{bmatrix}, \quad w = \begin{bmatrix}
w_1 \\
w_2 \\
\vdots \\
w_n
\end{bmatrix}, \quad \text{and} \quad e = \begin{bmatrix}
0 \\
0 \\
\vdots \\
1
\end{bmatrix}.
\]
Finally, since $A^*$ is the nonsingular matrix, the local weight vector can be derived as:

$$w = (A^*)^{-1} e,$$

(7)

where $(A^*)^{-1}$ is the inverse of $A^*$ and $A^*(A^*)^{-1} = I$. Next, we give an example to show how a local weight vector can be derived using Cogger and Yu’s method.

Let a positive upper triangular comparison matrix and diagonal matrix $D$ can be represented, respectively, as

$$A = \begin{bmatrix} 1 & 3 & 5 \\ 0 & 1 & 2 \\ 0 & 0 & 1 \end{bmatrix} \quad \text{and} \quad D = \begin{bmatrix} 3 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$ 

Then, we can obtain

$$A^* = \begin{bmatrix} -2 & 3 & 5 \\ 0 & -1 & 2 \\ 1 & 1 & 1 \end{bmatrix}.$$

The weight vector can be derived as:

$$w = \begin{bmatrix} -0.1765 & 0.1176 & 0.6471 \\ 0.1176 & -0.4118 & 0.2353 \\ 0.0588 & 0.2941 & 0.1176 \end{bmatrix} = \begin{bmatrix} 0.6471 \\ 0.2353 \\ 0.1176 \end{bmatrix},$$

where

$$(A^*)^{-1} = \begin{bmatrix} -0.1765 & 0.1176 & 0.6471 \\ 0.1176 & -0.4118 & 0.2353 \\ 0.0588 & 0.2941 & 0.1176 \end{bmatrix}.$$ 

Therefore, the work of finding weights in the AHP is transformed to calculate the last column vector of the inverse of the matrix $A^*$.

Once we derive all local weight vectors in the AHP, we can form the supermatrix according to the particular network structure. The general form of a supermatrix can be represented as follows:
where $C_m$ denotes the $m$th cluster, $e_{mn}$ denotes the $n$th element in the $m$th cluster, and $W_{ij}$ is the local priority matrix in which the $j$th cluster influences the $i$th cluster. Note that if the $j$th cluster has no influence on the $i$th cluster, then $W_{ij} = 0$.

A simple case of a network structure is given in Fig. 1.

Fig. 1. A network structure.
According to Fig. 1, we can form the supermatrix as:

\[
\Pi = \begin{bmatrix}
  C_1 & C_2 & C_3 \\
  a_1 & a_2 & b_1 & b_2 & c_1 & c_2 \\
  C_1 & a_1 & b_1 & a_2 & b_2 & C_3 & c_1 & c_2 \\
  W_1 & W_2 & W_3 & 0 \\
  b_1 & b_2 & W_1 & W_2 & W_3 & 0 \\
  C_3 & c_1 & c_2 & 0 & W_1 & W_2 & W_3 & 0
\end{bmatrix},
\]

where \( W_{ij} = \begin{bmatrix} w_{ij1}, w_{ij2} \end{bmatrix} \)

Note that the local weight vectors \( w_{12} = [w_{121}, w_{122}] \) and \( w_{13} = [w_{131}, w_{132}] \) are derived by the AHP.

Then, the weighted supermatrix can be obtained by transforming the sum of all the columns equal unity exactly. This step is very similar to the concept of Markov chains, which requires that the sum of the probabilities of all states equals to one.

Next, for simplicity, we rewrite the general form of the supermatrix with the following matrix:

\[
\Pi = \begin{bmatrix}
  \pi_{11} & \pi_{12} & \cdots & \pi_{1m} \\
  \pi_{21} & \pi_{22} & \cdots & \pi_{2m} \\
  \vdots & \vdots & \ddots & \vdots \\
  \pi_{m1} & \pi_{m2} & \cdots & \pi_{mm}
\end{bmatrix},
\]

where \( \sum_{i=1}^{m} \pi_{ij} = 1, \pi_{ij} \geq 0, \forall j = 1, \ldots, m. \)

Since \( \Pi \) can be viewed as a transition matrix of a Markov chain and \( \Pi^{(m)} \) has every entry positive (i.e., \( \Pi \) is regular), there is a unique column matrix \( \pi \) satisfying \( \Pi \pi = \pi \), and the entries of \( \pi \) are positive and sum to 1, where \( \pi \) can be regarded as the global weight vector in the ANP. Therefore, to derive the steady-state process of a supermatrix, we can solve the following system of linear equations:

\[
\begin{align*}
\pi_1 &= \pi_1 \pi_1 + \pi_2 \pi_2 + \cdots + \pi_m \pi_m, \\
\pi_2 &= \pi_2 \pi_1 + \pi_2 \pi_2 + \cdots + \pi_m \pi_m, \\
&\vdots \\
\pi_m &= \pi_m \pi_1 + \pi_m \pi_2 + \cdots + \pi_m \pi_m.
\end{align*}
\]

(8)

By moving the right-side of Eq. (8) to the left-side, we can rewrite Eq. (8) as:
A Matrix Method for the Fuzzy Analytic Network Process

\[(1 - \pi_{11})\pi_{11} - \pi_{12}\pi_{22} - \ldots - \pi_{1n}\pi_{nn} = 0,\]
\[-\pi_{21}\pi_{11} + (1 - \pi_{22})\pi_{22} - \ldots - \pi_{2n}\pi_{nn} = 0,\]
\[\vdots\]
\[-\pi_{n1}\pi_{11} - \pi_{n2}\pi_{22} - \ldots + (1 - \pi_{nn})\pi_{nn} = 0.\]

(9)

Since the last equation of the above linear system is superfluous, we replace it with the constraint \(\pi'_1 = 1\), where \(1' = (1, 1, \ldots, 1)\). Then, Eq. (9) can be represented as the following matrix form:

\[B'\pi = e,\]

(10)

where

\[B' = \begin{bmatrix}
-\pi_{11} & -\pi_{12} & \cdots & -\pi_{1n} \\
-\pi_{21} & 1 - \pi_{22} & \cdots & -\pi_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
1 & 1 & \cdots & 1
\end{bmatrix}, \pi = \begin{bmatrix}
\pi_1 \\
\pi_2 \\
\vdots \\
\pi_n
\end{bmatrix}, \text{ and } e = \begin{bmatrix}
0 \\
0 \\
\vdots \\
1
\end{bmatrix}.

Finally, if \(B'\) is the nonsingular matrix, the global weight vector can be derived as

\[\pi = (B')^{-1}e,\]

(11)

where \((B')^{-1}\) denotes the inverse of \(B'\) and \(B' (B')^{-1} = I\).

Similar to the result of the AHP, the work of finding weights of criteria in the ANP is transformed to calculate the last column vector of the inverse of the matrix \(B'\). In order to demonstrate the proposed method, we give an example as follows.

Let a supermatrix be formed as:

\[\Pi = \begin{bmatrix}
0.1290 & 0.6223 & 0.5171 & 0.0657 \\
0.6066 & 0.0000 & 0.1243 & 0.2146 \\
0.1984 & 0.1307 & 0.0000 & 0.1869 \\
0.0660 & 0.2470 & 0.3586 & 0.5327
\end{bmatrix}.

Then, we can obtain

\[B' = \begin{bmatrix}
1 - 0.1290 & -0.6223 & -0.5171 & -0.0657 \\
-0.6066 & 1 - 0.0000 & -0.1243 & -0.2146 \\
-0.1984 & -0.1307 & 1 - 0.0000 & -0.1869 \\
1 & 1 & 1 & 1
\end{bmatrix}.

Finally, we can derive the global weight vector as

\[\pi = \begin{bmatrix}
1.3169 & 0.5824 & 0.4565 & 0.2968 \\
0.4256 & 1.0144 & 0.0847 & 0.2615 \\
-0.0074 & -0.0424 & 0.8429 & 0.1480 \\
-1.7351 & -1.5544 & -1.3841 & 0.2937
\end{bmatrix} \begin{bmatrix}
0 \\
0 \\
0 \\
1
\end{bmatrix} = \begin{bmatrix}
0.2968 \\
0.2615 \\
0.1480 \\
0.2937
\end{bmatrix},

where
We can summarize the characteristics of the proposed method as follows. First, the proposed method does not need to hold the property of the reciprocal matrix in the AHP. The first property makes the possible way to naturally extend the AHP to the FAHP. Second, instead of solving the limiting power of a supermatrix, a global weight vector can be derived by solving the particular matrix problem. The second property avoids the convergent problem of the fuzzy supermatrix. Next, we will describe how to derive the procedures of the FANP in Section 3.

3. The Fuzzy Analytic Network Process

In order to consider the ANP under fuzzy environments, fuzzy numbers are used to compare the ratio of weights between criteria. In this paper, a fuzzy number is presented as the triangular form. Other forms of fuzzy numbers can be easily employed using the same procedures.

Let a fuzzy positive upper triangular comparison matrix

$$A = [a_{ij}]_{n \times n}, \quad a_{ij} = \begin{cases} \tilde{a}_{ij} & \text{if } i \leq j, \\ 0 & \text{otherwise,} \end{cases}$$

(12)

where $\tilde{a}_{ij}$ denotes the strength of the ratio of the weight that the criterion $i$ dominates the criterion $j$.

Then, the matrix of $A^r$ can be represented as

$$A^r = \begin{bmatrix} 1-n & \tilde{a}_{12} & \cdots & \tilde{a}_{1s-1} & \tilde{a}_{1s} \\ 0 & 2-n & \cdots & \tilde{a}_{2s-1} & \tilde{a}_{2s} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & -1 & \tilde{a}_{s-1s} \\ 1 & 1 & \cdots & 1 & 1 \end{bmatrix}.$$  

(13)

It is clear that if we can derive the inverse of $A^r$, we can obtain local fuzzy weights in the AHP. Therefore, to find the inverse of a fuzzy matrix, at least two methods can be used: the linear programming approach and Cramer’s rule. Next, we briefly introduce the above methods as follows.

Let us first consider the crisp case and $C$ be a nonsingular matrix. The inverse of $C$, denoted by $C^{-1}$, holds the following property

$$C \ast C^{-1} = I,$$

(14)
or
A Matrix Method for the Fuzzy Analytic Network Process

\[
\begin{pmatrix}
  c_{11} & \cdots & c_{1n} \\
  \vdots & \ddots & \vdots \\
  c_{n1} & \cdots & c_{nn}
\end{pmatrix}
\begin{pmatrix}
  c'_{11} & \cdots & c'_{1n} \\
  \vdots & \ddots & \vdots \\
  c'_{n1} & \cdots & c'_{nn}
\end{pmatrix} =
\begin{pmatrix}
  1 & \cdots & 1 \\
  \vdots & \ddots & \vdots \\
  1 & \cdots & 1
\end{pmatrix}.
\]

(15)

To derive \( C^{-1} \), we can rewrite Eq. (15) and solve the following system of linear equations synchronously:

\[
\begin{align*}
& c_{11}c'_{11} + c_{12}c'_{21} + \cdots + c_{1n}c'_{n1} = 1, \\
& c_{11}c'_{12} + c_{12}c'_{22} + \cdots + c_{1n}c'_{n2} = 1, \\
& \vdots \phantom{= 1}, \\
& c_{11}c'_{1n} + c_{12}c'_{2n} + \cdots + c_{1n}c'_{nn} = 1,
\end{align*}
\]

(16)

where \( I \) denotes the identity matrix. For example to derive the last column vector of \( C^{-1} \), we can solve the following linear system:

\[
\begin{align*}
& c_{11}c'_{11} + c_{12}c'_{21} + \cdots + c_{1n}c'_{n1} = 1, \\
& c_{21}c'_{11} + c_{22}c'_{21} + \cdots + c_{2n}c'_{n1} = 1, \\
& \vdots \phantom{= 1}, \\
& c_{n1}c'_{11} + c_{n2}c'_{21} + \cdots + c_{nn}c'_{n1} = 1,
\end{align*}
\]

(17)

In addition, we can also solve the Eq. (17) using Cramer’s rule such that

\[ c'_{jn} = \frac{|C_j|}{|C|}, \]

(18)

where \( C_j \) is \( C \) with its \( j \)th column replaced by \( \mathbf{1'} = (1, 1, \ldots, 1) \). Other column vectors of \( C^{-1} \) can be derived using the same procedure.

For fuzzy numbers, we can solve the following linear programming problem for deriving local fuzzy weights in the AHP:

\[
\text{max/min } c'_{ij}
\]

s.t. \( c_{ij}c'_{ij} + c_{ij}c'_{ij} + \cdots + c_{ij}c'_{ij} = 1, \)
\( c_{ij}c'_{ij} + c_{ij}c'_{ij} + \cdots + c_{ij}c'_{ij} = 1, \)
\( \vdots \phantom{= 1}, \)
\( c_{ij}c'_{ij} + c_{ij}c'_{ij} + \cdots + c_{ij}c'_{ij} = 1, \)
\( c'_{ij} \in C_j[\alpha], \quad c_{ij} \in [0,1], \forall i, j = 1, \ldots, n, \)

where \( C_j[\alpha] = [\min c'_j, \max c'_j] \) denotes the fuzzy element and \( [\alpha] \) is the \( \alpha \) -cut operation.

Or, by using Cramer’s rule, we can directly fuzzify Eq. (18) to derive the inverse of a fuzzy matrix. For example to derive \( \tilde{c}'_{ij}[\alpha] = \{c'_{ij}(\alpha), c'_{ij}(\alpha)\} \), we can calculate

\[ c'_{ij}(\alpha) = \min \left\{ \frac{|C_j|}{|C|} \mid c \in \tilde{c}[\alpha] \right\}, \]

(20)
and
\[ c'_{\max}(\alpha) = \max \left\{ \frac{|C_c|}{|C|} \mid c \in \tilde{c}[\alpha] \right\}. \tag{21} \]

Since Eqs. (19)-(21) may be hard to evaluate, some heuristic algorithms (e.g., genetic algorithm, ant algorithm, or simulated annealing) can be used to obtain approximate solutions.

Next, an example is given to show the procedure that how the inverse of a fuzzy matrix can be derived in the AHP. Assume that the upper triangular fuzzy comparison matrix can be given by the decision-maker as:

\[
\begin{bmatrix}
1 & (2,3,4) & (1/6,1/5,1/4) & (6,7,8) \\
0 & 1 & (1/8,1/7,1/6) & (1,2,3) \\
0 & 0 & 1 & (8,9,9) \\
0 & 0 & 0 & 1
\end{bmatrix}
\]

Then, \( \tilde{A}^\star \) can be formed as:

\[
\begin{bmatrix}
-3 & (2,3,4) & (1/6,1/5,1/4) & (6,7,8) \\
0 & -2 & (1/8,1/7,1/6) & (1,2,3) \\
0 & 0 & -1 & (8,9,9) \\
1 & 1 & 1 & 1
\end{bmatrix}
\]

Next, by solving Eq. (19) and set the \( \alpha \)-cuts = 0, 0.2, 0.4, 0.6, 0.8, and 1, we can derive the fuzzy local weights, i.e., the last column vector of \( (\tilde{A}^\star)^{-1} \), as shown in Table 1.

<table>
<thead>
<tr>
<th>( \alpha )-cut</th>
<th>( \hat{w}_1[\alpha] )</th>
<th>( \hat{w}_2[\alpha] )</th>
<th>( \hat{w}_3[\alpha] )</th>
<th>( \hat{w}_4[\alpha] )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>[0.2248,0.3578]</td>
<td>[0.0668,0.1439]</td>
<td>[0.4601,0.6307]</td>
<td>[0.0536,0.0763]</td>
</tr>
<tr>
<td>0.2</td>
<td>[0.2359,0.3426]</td>
<td>[0.0737,0.1347]</td>
<td>[0.4786,0.6152]</td>
<td>[0.0551,0.0730]</td>
</tr>
<tr>
<td>0.4</td>
<td>[0.2472,0.3274]</td>
<td>[0.0806,0.1259]</td>
<td>[0.4974,0.5998]</td>
<td>[0.0567,0.0699]</td>
</tr>
<tr>
<td>0.6</td>
<td>[0.2587,0.3122]</td>
<td>[0.0874,0.1174]</td>
<td>[0.5164,0.5847]</td>
<td>[0.0583,0.0670]</td>
</tr>
<tr>
<td>0.8</td>
<td>[0.2704,0.2971]</td>
<td>[0.0943,0.1092]</td>
<td>[0.5356,0.5697]</td>
<td>[0.0600,0.0643]</td>
</tr>
<tr>
<td>1.0</td>
<td>[0.2821,0.2821]</td>
<td>[0.1013,0.1013]</td>
<td>[0.5549,0.5549]</td>
<td>[0.0617,0.0617]</td>
</tr>
</tbody>
</table>

To check the fuzzy local weights visually, we can depict the triangular-shaped fuzzy local weights of the given example as shown in Fig. 2.
On the other hand, to find the global weight vector in the FANP can also be viewed as the problem of calculating the inverse of a fuzzy matrix i.e., to solve $(B^*)^T$. Therefore, the method of finding global weights in the FANP is similar to the above procedures. Next, an application is used to demonstrate the proposed method in Section 4.

4. An Application

Consider the market share of a food company can be evaluated by three clusters: Advertising ability ($C_1$), Quality ability ($C_2$), and Attraction ability ($C_3$). Each cluster can be divided into three criteria, including Creativity, Promotion, Frequency, Nutrition, Taste, Cleanliness, Price, Location, and Reputation, respectively. The decision maker wants to determine the weights of the criteria using the ANP so that he/she can allocate the appropriate budgets for obtaining the maximum market share. Due to the restrictions of incomplete information and human subjective judgments, the decision-maker employs fuzzy numbers to judge the ratios of the weights between the criteria. The network structure adopted in this application to deal with the problem of the market share is depicted as shown in Fig. 3.
In order to calculate all the fuzzy local vectors, we should first compare the relative fuzzy ratios of the weights between the criteria, and then the corresponding fuzzy local weights with $\alpha$-cut = 0 can be derived using Eq. (19). The results can be given as shown in the following nine matrices:

<table>
<thead>
<tr>
<th>Creativity</th>
<th>Nutrition</th>
<th>Taste</th>
<th>Cleanliness</th>
<th>Fuzzy local vector</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nutrition</td>
<td>1</td>
<td>(1/3,1/2,1)</td>
<td>(3,4,5)</td>
<td>(0.2632,0.3514,0.4737)</td>
</tr>
<tr>
<td>Taste</td>
<td>0</td>
<td>1</td>
<td>(4,5,6)</td>
<td>(0.4211,0.5405,0.6316)</td>
</tr>
<tr>
<td>Cleanliness</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>(0.0800,0.1081,0.1395)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Promotion</th>
<th>Nutrition</th>
<th>Taste</th>
<th>Cleanliness</th>
<th>Fuzzy local weights</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nutrition</td>
<td>1</td>
<td>(3,4,5)</td>
<td>(4,5,6)</td>
<td>(0.6250,0.6842,0.7333)</td>
</tr>
<tr>
<td>Taste</td>
<td>0</td>
<td>1</td>
<td>(1,2,2)</td>
<td>(0.1333,0.2105,0.2500)</td>
</tr>
<tr>
<td>Cleanliness</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>(0.0909,0.1053,0.1818)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Frequency</th>
<th>Nutrition</th>
<th>Taste</th>
<th>Cleanliness</th>
<th>Fuzzy local vector</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nutrition</td>
<td>1</td>
<td>(1/3,1/2,1)</td>
<td>(1/3,1/2,1)</td>
<td>(0.1429,0.2000,0.3333)</td>
</tr>
<tr>
<td>Taste</td>
<td>0</td>
<td>1</td>
<td>(1/2,1,1)</td>
<td>(0.2222,0.4000,0.4286)</td>
</tr>
<tr>
<td>Cleanliness</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>(0.3333,0.4000,0.5714)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Frequency</th>
<th>Price</th>
<th>Location</th>
<th>Reputation</th>
<th>Fuzzy local vector</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nutrition</td>
<td>1</td>
<td>(1,2,3)</td>
<td>(1/5,1/4,1/3)</td>
<td>(0.1358,0.2131,0.3023)</td>
</tr>
<tr>
<td>Location</td>
<td>0</td>
<td>1</td>
<td>(1/6,1/5,1/4)</td>
<td>(0.1053,0.1312,0.1695)</td>
</tr>
<tr>
<td>Reputation</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>(0.5581,0.6557,0.7407)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Taste</th>
<th>Price</th>
<th>Location</th>
<th>Reputation</th>
<th>Fuzzy local vector</th>
</tr>
</thead>
<tbody>
<tr>
<td>Price</td>
<td>1</td>
<td>(2,3,4)</td>
<td>(1/4,1/3,1/2)</td>
<td>(0.2000,0.2800,0.3750)</td>
</tr>
<tr>
<td>Location</td>
<td>0</td>
<td>1</td>
<td>(1/6,1/5,1/4)</td>
<td>(0.0952,0.1200,0.1538)</td>
</tr>
<tr>
<td>Reputation</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>(0.5000,0.6000,0.6857)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Cleanliness</th>
<th>Price</th>
<th>Location</th>
<th>Reputation</th>
<th>Fuzzy local vector</th>
</tr>
</thead>
<tbody>
<tr>
<td>Price</td>
<td>1</td>
<td>(3,4,5)</td>
<td>(1/3,1/2,1)</td>
<td>(0.2800,0.3750,0.5000)</td>
</tr>
<tr>
<td>Location</td>
<td>0</td>
<td>1</td>
<td>(1/5,1/4,1/3)</td>
<td>(0.0909,0.1250,0.1667)</td>
</tr>
<tr>
<td>Reputation</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>(0.3750,0.5000,0.6000)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Price</th>
<th>Creativity</th>
<th>Promotion</th>
<th>Frequency</th>
<th>Fuzzy local vector</th>
</tr>
</thead>
<tbody>
<tr>
<td>Creativity</td>
<td>1</td>
<td>(1/4,1/3,1/2)</td>
<td>(2,3,4)</td>
<td>(0.2000,0.2800,0.3750)</td>
</tr>
<tr>
<td>Promotion</td>
<td>0</td>
<td>1</td>
<td>(4,5,6)</td>
<td>(0.5000,0.6000,0.6857)</td>
</tr>
<tr>
<td>Frequency</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>(0.0952,0.1200,0.1538)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Location</th>
<th>Creativity</th>
<th>Promotion</th>
<th>Frequency</th>
<th>Fuzzy local vector</th>
</tr>
</thead>
<tbody>
<tr>
<td>Creativity</td>
<td>1</td>
<td>(1/5,1/4,1/3)</td>
<td>(1/4,1/3,1/2)</td>
<td>(0.0960,0.1220,0.1724)</td>
</tr>
<tr>
<td>Promotion</td>
<td>0</td>
<td>1</td>
<td>(1/2,3)</td>
<td>(0.4138,0.5853,0.6779)</td>
</tr>
<tr>
<td>Frequency</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>(0.2105,0.2927,0.4494)</td>
</tr>
</tbody>
</table>
Then, we can formulate the fuzzy supermatrix as:

\[
\tilde{\Pi} = \begin{bmatrix}
    0 & \tilde{w}_{12} & \tilde{w}_{13} \\
    \tilde{w}_{21} & 0 & \tilde{w}_{23} \\
    0 & \tilde{w}_{31} & 0
\end{bmatrix}, \quad \text{where} \quad \tilde{0} = \begin{bmatrix}
    0 & 0 & 0
\end{bmatrix},
\]

\[
\tilde{w}_{13} = \begin{bmatrix}
    0.2000, 0.2800, 0.3750 \\
    0.0960, 0.1220, 0.1724 \\
    0.6154, 0.6800, 0.7273
\end{bmatrix},
\]

\[
\tilde{w}_{21} = \begin{bmatrix}
    0.5000, 0.6000, 0.6857 \\
    0.4138, 0.5853, 0.6779 \\
    0.1818, 0.2400, 0.3077
\end{bmatrix},
\]

\[
\tilde{w}_{23} = \begin{bmatrix}
    0.0952, 0.1200, 0.1538 \\
    0.2105, 0.2927, 0.4494 \\
    0.1429, 0.2000, 0.3333
\end{bmatrix},
\]

\[
\tilde{w}_{32} = \begin{bmatrix}
    0.2632, 0.3514, 0.4737 \\
    0.6250, 0.6842, 0.7333 \\
    0.1429, 0.2000, 0.3333
\end{bmatrix},
\]

Let \( \alpha \)-cuts = 0, 0.2, 0.4, 0.6, 0.8, and 1.0, and we can obtain the fuzzy global weights by calculating \((\tilde{B})^{-1}\) as shown in Table 2.
Table 2. The fuzzy global weights in the application.

<table>
<thead>
<tr>
<th>Fuzzy global weights</th>
<th>$\alpha$ - cut = 0</th>
<th>$\alpha$ - cut = 0.2</th>
<th>$\alpha$ - cut = 0.4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Creativity</td>
<td>[0.1257,0.2047]</td>
<td>[0.1345,0.1976]</td>
<td>[0.1433,0.1904]</td>
</tr>
<tr>
<td>Promotion</td>
<td>[0.0860,0.1700]</td>
<td>[0.0936,0.1590]</td>
<td>[0.1014,0.1484]</td>
</tr>
<tr>
<td>Frequency</td>
<td>[0.0250,0.0651]</td>
<td>[0.0277,0.0596]</td>
<td>[0.0305,0.0542]</td>
</tr>
<tr>
<td>Nutrition</td>
<td>[0.1106,0.2005]</td>
<td>[0.1190,0.1905]</td>
<td>[0.1275,0.1805]</td>
</tr>
<tr>
<td>Taste</td>
<td>[0.0840,0.1704]</td>
<td>[0.0934,0.1628]</td>
<td>[0.1030,0.1552]</td>
</tr>
<tr>
<td>Cleanliness</td>
<td>[0.0323,0.0829]</td>
<td>[0.0351,0.0748]</td>
<td>[0.0380,0.0672]</td>
</tr>
<tr>
<td>Price</td>
<td>[0.0547,0.1310]</td>
<td>[0.0611,0.1218]</td>
<td>[0.0676,0.1129]</td>
</tr>
<tr>
<td>Location</td>
<td>[0.0289,0.0613]</td>
<td>[0.0313,0.0571]</td>
<td>[0.0339,0.0531]</td>
</tr>
<tr>
<td>Reputation</td>
<td>[0.1549,0.2493]</td>
<td>[0.1645,0.2400]</td>
<td>[0.1742,0.2307]</td>
</tr>
<tr>
<td></td>
<td>$\alpha$ - cut = 0.6</td>
<td>$\alpha$ - cut = 0.8</td>
<td>$\alpha$ - cut = 1.0</td>
</tr>
<tr>
<td>Creativity</td>
<td>[0.1521,0.1831]</td>
<td>[0.1602,0.1757]</td>
<td>[0.1682,0.1682]</td>
</tr>
<tr>
<td>Promotion</td>
<td>[0.1094,0.1381]</td>
<td>[0.1177,0.1321]</td>
<td>[0.1260,0.1260]</td>
</tr>
<tr>
<td>Frequency</td>
<td>[0.0334,0.0490]</td>
<td>[0.0363,0.0441]</td>
<td>[0.0391,0.0391]</td>
</tr>
<tr>
<td>Nutrition</td>
<td>[0.1360,0.1704]</td>
<td>[0.1446,0.1618]</td>
<td>[0.1532,0.1532]</td>
</tr>
<tr>
<td>Taste</td>
<td>[0.1128,0.1478]</td>
<td>[0.1230,0.1405]</td>
<td>[0.1331,0.1331]</td>
</tr>
<tr>
<td>Cleanliness</td>
<td>[0.0409,0.0601]</td>
<td>[0.0440,0.0536]</td>
<td>[0.0471,0.0471]</td>
</tr>
<tr>
<td>Price</td>
<td>[0.0742,0.1042]</td>
<td>[0.0809,0.0959]</td>
<td>[0.0876,0.0876]</td>
</tr>
<tr>
<td>Location</td>
<td>[0.0365,0.0492]</td>
<td>[0.0393,0.0456]</td>
<td>[0.0420,0.0420]</td>
</tr>
<tr>
<td>Reputation</td>
<td>[0.1841,0.2216]</td>
<td>[0.1940,0.2127]</td>
<td>[0.2038,0.2038]</td>
</tr>
</tbody>
</table>

Next, to show the justification of the proposed method, we find the crisp global weights using the vertices of the fuzzy numbers in the fuzzy supermatrix, and show they belong to the alpha-zero cut of the fuzzy global weights.

Let a crisp supermatrix

$$\Pi = \begin{bmatrix} 0 & 0 & W_{23} \\ W_{21} & 0 & 0 \\ 0 & W_{32} & 0 \end{bmatrix},$$

where $0 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$, $W_{23} = \begin{bmatrix} 0.3750 & 0.1724 & 0.7273 \\ 0.5000 & 0.4138 & 0.1818 \\ 0.1250 & 0.4138 & 0.0909 \end{bmatrix}$, $W_{21} = \begin{bmatrix} 0.4737 & 0.7333 & 0.3333 \\ 0.4211 & 0.1333 & 0.2222 \\ 0.1052 & 0.1334 & 0.4445 \end{bmatrix}$, and $W_{32} = \begin{bmatrix} 0.3023 & 0.3750 & 0.5000 \\ 0.1053 & 0.0952 & 0.0909 \\ 0.5924 & 0.5298 & 0.4091 \end{bmatrix}$.

By calculating the steady-state process of $\Pi$, the global weight vector can be derived by raising $\Pi$ to limiting power as:

$$\pi = \begin{bmatrix} 0.1821 & 0.1061 & 0.0451 & 0.1791 & 0.1009 & 0.0534 & 0.1187 & 0.0333 & 0.1814 \end{bmatrix}.$$
Clearly, it belongs to the alpha-zero cut of the fuzzy global weight vector. Readers can use other vertices of the fuzzy numbers to show all crisp global weights belong to the alpha-zero cut of the fuzzy global weights.

Hence, we can conclude that the proposed method can obtain the appropriate fuzzy global weights and overcome the convergent problem of a fuzzy limiting supermatrix. However, we should highlight the possible difficulty that the decision maker may have in allocating the budgets based on the results of the fuzzy global weights. Clearly, since the global weights of the criteria are fuzzy numbers, it is not straightforward for a decision-maker to make the decision. In this situation, the problem of ranking fuzzy numbers is considered to compare fuzzy numbers.

Many defuzzification methods have been proposed to rank fuzzy numbers, including preference relation methods, fuzzy mean and spread methods, fuzzy scoring method, and linguistic expression methods. One of them can easily be employed to obtain crisp overall scores of alternatives. However, it should be highlighted that none of all defuzzification methods can outperform to others in all fuzzy situations. In addition, since the purpose of this paper is to propose extend the ANP under fuzzy environments, the issue of ranking fuzzy numbers is ignored. Next, we provide the depth discussions according to the finding of our application in Section 5.

5. Discussions

The ANP is a popular technique used to model subjective decision-making processes according to multiple dependent and conflict criteria. Recently, it has been widely used in corporate planning, portfolio selection, and benefit/cost analysis by government agencies for resource allocation purposes. However, a crucial issue about the ANP under fuzzy environments seems to be ignored. Since fuzzy global weights may provide the important information for decision-makers to make dynamic decisions, we should develop an appropriate approach to provide the information of fuzzy global weights in the ANP.

Two problems for extending the ANP under fuzzy environments are considered and overcome in this paper. First, the problematic postulate that $\bar{a}_{ij} = (\bar{a}_{ij})^{-1}$ is released by directly fuzzifying Cogger and Yu’s eigenvector method. Second, the convergent problem of a fuzzy limiting supermatrix is overcome by solving the particular matrix problem. From the results of the application, it can be seen that the proposed method can fully extend the ANP to consider fuzzy situations.

Compared to the crisp ANP, the advantages of the proposed method can be described as follows. First, due to the restrictions of incomplete information and subjective uncertainty, fuzzy numbers are more suitable for judging the relative weights between criteria. Furthermore, the fuzzy global weights can be obtained for decision-makers to understand the degrees of uncertainty. It should be highlighted that the ANP is one special case of the proposed method when $\alpha$ - cut = 1.

6. Conclusions

Although the ANP has been widely used in various applications, it is hard for a decision-maker to quantify precise ratios of weights between criteria under incomplete information and subjective uncertainty. In this paper, the FANP is proposed to extend the conventional ANP and the fuzzy judgments are used to compare the relative ratios of
weights between criteria. For obtaining the appropriate fuzzy local/global weights, we first derive the AHP/ANP problem to the particular matrix problem. Then, we solve the inverse of the fuzzy matrix to obtain the fuzzy local/global weights. Furthermore, from the results of our application, we can conclude that the proposed method can appropriately extend the ANP under fuzzy environments.

Acknowledgements

This paper is supported by National Science Council of Taiwan under the Grant NSC-96-2416-H-424-003.

References