

## Research Article

# Repairing the Inconsistent Fuzzy Preference Matrix Using Multiobjective PSO

Abba Suganda Girsang,<sup>1</sup> Chun-Wei Tsai,<sup>2</sup> and Chu-Sing Yang<sup>3</sup>

<sup>1</sup>Master of Information Technology at Binus Graduate Program, Bina Nusantara University, Jalan Kebon Jeruk Raya No. 27, Jakarta 11530, Indonesia

<sup>2</sup>Department of Computer Science and Information Engineering, National Ilan University, Yilan 26041, Taiwan

<sup>3</sup>Institute of Computer and Communication Engineering and Department of Electrical Engineering, National Cheng Kung University, Tainan 70101, Taiwan

Correspondence should be addressed to Abba Suganda Girsang; [gandagirsang@yahoo.com](mailto:gandagirsang@yahoo.com)

Received 26 August 2015; Accepted 8 October 2015

Academic Editor: Katsuhiro Honda

Copyright © 2015 Abba Suganda Girsang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper presents a method using multiobjective particle swarm optimization (PSO) approach to improve the consistency matrix in analytic hierarchy process (AHP), called PSOMOF. The purpose of this method is to optimize two objectives which conflict each other, while improving the consistency matrix. They are minimizing consistent ratio (CR) and deviation matrix. This study focuses on fuzzy preference matrix as one model comparison matrix in AHP. Some inconsistent matrices are repaired successfully to be consistent by this method. This proposed method offers some alternative consistent matrices as solutions.

## 1. Introduction

One important issue in comparison matrix of AHP is the consistency. In multicriteria decision making (MCDM), decision makers (DMs) reveal their opinion to choose some decision alternatives by a comparison matrix [1]. However, the comparison matrix which is identified as inconsistent cannot be used as a judgment. Meanwhile, the consistency is hard to obtain, when evaluating a large number of criteria.

There are two models of a comparison matrix, multiplicative preference relations [1] and fuzzy preference relations [2, 3]. The element comparison matrix of multiplicative preference relation is stated as  $a_{ij}$  which defines the dominance of alternative  $i$  over  $j$ , where  $1 < a_{ij} < 9$  and  $a_{ij} = 1/a_{ji}$ . On fuzzy preference relations, element comparison matrix is stated as  $a_{ij}$ , which defines the preference of alternative  $i$  over  $j$ , where  $0 < a_{ij} < 1$  and  $a_{ij} + a_{ji} = 1$ . This study focuses on fuzzy preference relations.

The issues of consistency in fuzzy preference relation also have received attention from researchers. Xu and Wang [4] proposed a revised approach by using linear programming models to generate the priority weights for additive interval

fuzzy preference relations. Xu and Chen [5] presented the method to fulfill the element, which is incomplete on fuzzy preference for group decision making based on additive transitive consistency and accumulates the auxiliary value into a group auxiliary relation. This research was extended by Xu et al. [6], who deduced a function between the additive transitivity fuzzy preference and its corresponding priority vector. Xu et al. [7] proposed algorithm by eliminating the cycles of length 3 to  $n$  in the digraph of the incomplete reciprocal preference relation and converted it to the one with ordinal consistency. Liu et al. [8] proposed a method to solve the incompleteness of fuzzy preference matrix and also repair the inconsistency preference matrix. This method calculated minimal of the squared error of the incomplete fuzzy preference relation and its priority weight vector to fulfill the missing values and generated the consistency fuzzy preference such that the modified one is the closest to the original one. Chen et al. [9] presented a method for group decision making using incomplete fuzzy preference based on additive consistency. Chiclana et al. [10] proposed a functional equation to model the cardinal consistency in the strength of preferences of reciprocal preference relations.

Xia et al. [11] improved the consistency by using the geometric consistency index in complete and incomplete fuzzy preference.

A research using swarm intelligence was also used to solve the inconsistent comparison matrix such as PSO which combines Taguchi method [12]. It improved the previous research using genetic algorithm [13] to solve the inconsistent comparison matrix. Both researches used the same objective function to solve the problem, that is, summing the CR and deviation matrix. Although successful metaheuristic to solve that problem, the variations of implemented metaheuristic, is rarely conducted. Girsang et al. [14, 15] also already implemented the ant colony optimization (ACO) approach in our previous research to solve this problem with the different objective function that uses Yang et al. [12] and Lin et al. [13]. In [14], besides repairing the inconsistent ratio, ACO is used to enhance the minimal deviation matrix, while in [15] ACO is used to enhance the minimal consistent ratio. It becomes a promising research to consider both of the two objective functions using swarm intelligence. Girsang et al. [16] also implemented PSO with multiobjective approach; however, it only focuses on repairing the multiplicative preference matrix.

## 2. Related Work

*2.1. Consistent Ratio in AHP.* A simple illustration about inconsistency is described as follows. The decision maker (DM) has opinion that  $X$  is bigger than  $Y$  and  $Y$  is bigger than  $Z$ . The consistent logic of this case is that  $X$  should be bigger than  $Z$ . Contrarily, it would be inconsistent if DM said that  $Z$  is bigger than  $X$ . In AHP, the opinion of decision makers is represented in a comparison matrix. An element comparison matrix can reflect the subjective opinion that expose strength of the preference and the feeling. In a fuzzy preference matrix, the element of comparison matrix,  $A$ , can be expressed as  $a_{ij}$ , with a scale value ( $0 \cdots 1$ ) where  $0 < a_{ij} < 1$ ,  $a_{ij} + a_{ji} = 1$ , and  $a_{ii} = 0.5$ . Matrix  $A$  as Fuzzy preference relation can be depicted as follows:

$$A = \begin{pmatrix} 0.5 & 1 - a_{21} & 1 - a_{31} & 1 - a_{41} \\ a_{21} & 0.5 & 1 - a_{32} & 1 - a_{42} \\ a_{31} & a_{32} & 0.5 & 1 - a_{43} \\ a_{41} & a_{42} & a_{43} & 0.5 \end{pmatrix}. \quad (1)$$

To measure the multiplicative consistency in a comparison matrix, Saaty defined consistent ratio (CR). He proposed that the threshold of CR in multiplicative preference matrices is 0.1. The CR is defined as

$$AW = \lambda_{\max} W, \quad (2)$$

$$CI = \frac{\lambda_{\max} - n}{n - 1}, \quad (3)$$

$$CR = \frac{CI}{RI}, \quad (4)$$

where  $\lambda_{\max}$  and  $W$  are the eigenvalue and eigenvector of the matrix, respectively. Further, CI is the consistency index;

TABLE 1: Random consistency index (RI).

Number criteria	1	2	3	4	5	6	7	8	9
	0	0	0.58	0.9	1.12	1.24	1.32	1.41	1.45

$n$  represents number criteria or size matrix, and the RI (random consistency index) is the average index of randomly generated weights. The value of RI on each size matrices is described in Table 1. A CR less than 0.1 can be categorized as consistent matrix. Perfect consistency is obtained when the maximum eigenvalue equal to the number criteria ( $\lambda_{\max} = n$ ).

Herrera-Viedma et al. [17] proposed some definitions to reveal the consistency in a fuzzy preference matrix. They show that the additive consistency is more appropriate to define the degree of consistency of fuzzy preference matrix. The relation in matrix  $A$  is consistent if the element matrix can satisfy (5) and (6):

$$a_{ij} + a_{jk} + a_{ki} = \frac{3}{2}, \quad \forall i, j, k, \quad (5)$$

where

$$w_i = \frac{\sum_{j=1}^n a_{ij} - 0.5}{n(n-1)/2}. \quad (6)$$

Xu and Da [18] proposed determining the multiplicative consistency in the fuzzy preference matrix. They used Xu's [19] approach to determine CI in multiplicative preference matrix. Suppose  $b_{ij}$  is the element of multiplicative of preference matrix. Xu [19] defined the CI in (7). This equation is derived from (3). By knowing the weight of each criterion and element of matrix, CI can be obtained:

$$CI = \frac{1}{n(n-1)} \sum_{1 \leq i < j \leq n} \left[ b_{ij} \cdot \frac{w_j}{w_i} + b_{ji} \cdot \frac{w_i}{w_j} - 2 \right], \quad (7)$$

where  $b_{ij}$  is element matrix of multiplicative preference matrix and  $w = (w_1, w_2, w_3, \dots, w_n)$  is the weight of each criterion. To determine the CI in a fuzzy preference matrix, [4, 18] used converting with assumption  $b_{ij} = a_{ij}/a_{ji}$ , where  $a_{ij}$  is element matrix of the fuzzy preference matrix. Therefore, they proposed determining CI as in

$$CI = \frac{1}{n(n-1)} \sum_{1 \leq i < j \leq n} \left[ \frac{a_{ij}}{a_{ji}} \cdot \frac{w_j}{w_i} + \frac{a_{ji}}{a_{ij}} \cdot \frac{w_i}{w_j} - 2 \right]. \quad (8)$$

*2.2. Deviation Matrix.* While the consistent ratio is repaired, the modified matrix automatically generates the deviation matrix from the original. Ideally, the modified matrices are kept closer to their original matrices in order to maintain the original judgment. It means that the deviation matrix is enriched to be minimal. There are some methods to represent the deviation, such as difference index (Di) [13] and  $\delta$  and  $\sigma$ . Difference index (Di) is defined as the real difference between

the same gene values in two genotypes. The other deviation is defined as  $\delta$  and  $\sigma$ , which is denoted as

$$\delta = \max_{i,j} \{|a'_{i,j} - a_{i,j}|\}, \quad (9)$$

$$\sigma = \frac{\sqrt{\sum_{i=1}^n \sum_{j=1}^n (a'_{i,j} - a_{i,j})^2}}{n}, \quad (10)$$

where  $A$  is the original matrix  $[a_{ij}]$ ;  $A'$  is the modified matrix  $[a'_{i,j}]$ , and  $n$  is the matrix size.

In the multiplicative preference matrix, the difference index (Di) is generally used to measure the distance between two matrices. However, in fuzzy preference matrix,  $\delta$  and  $\sigma$  are considered as more appropriate to represent the deviation matrix. Since the value preference will be 0.51 to 1 or 0 to 0.49, the division each all genes in Di will not be different significantly. As a consequence, the difference of two matrices will not be significant as well. Therefore, in this study, instead of using Di,  $\sigma$  is employed to define the deviation matrix for the preference fuzzy matrix.

**2.3. Particle Swarm Optimization.** PSO was firstly proposed by Kennedy and Eberhart [20]. It is population-based stochastic optimization on the social behaviors observed in animals or insects such as bird flocking, fish schooling, and animal herding. In PSO, each particle of swarm represents the solution which moves to search the optimal solutions. Each particle also broadcasts its current position to neighbour particles. The position of each particle is adjusted according to its velocity and the best position it has found so far. A particle  $i$  starts moving with a velocity  $Vi(t + 1)$  from its current position  $Xi(t)$ , to the next position  $Xi(t + 1)$ , as in (11). The velocity is influenced by three factors: (a) previous velocity  $Vi(t)$ , (b) the best previous particle position  $Xp(t)$ , and (c) the best previous swarm particle position  $Xg(t)$ . It can be stated as (12):

$$Xi(t + 1) = Xi(t) + Vi(t + 1) \quad (11)$$

$$Vi(t + 1) = (w * Vi(t)) + (C_1 * R_1 (Xp(t) - Xi(t)) + (C_2 * R_2 (Xg(t) - Xi(t))), \quad (12)$$

where  $w$  is the weight to control the convergence of the velocity,  $C_1$  the acceleration weight cognitive element,  $C_2$  the weight of social parameter, and  $R_1$  and  $R_2$  are random numbers in the range  $[0, 1]$ .

### 3. Proposed Method

**3.1. PSOMOF Algorithm.** In PSO, each particle seeks the best position by moving in the search space. The position in PSO can represent an element in the comparison matrix. As shown in previous section, encoding position of element matrix can be encoded only from lower triangular matrix. If matrix  $A$  is identified as an inconsistent matrix and needs to be repaired, then the scale value of matrix should be changed with new value. To be efficient, the whole elements

of comparison matrix can be represented by lower triangular matrix. Therefore, the position of PSO that should be changed can be represented only by the lower triangular matrix. When changing the value of each node to be consistent, it also changes the rate consistent ratio (CR) and deviation matrix. We use  $\sigma$  to represent the deviation matrix in this method. Changing the value of each node means changing the particle's position. In PSO, the position is affected by a particles historical best position (local best) and the swarms' best position (global best). The solution (new value position) is performed to chase the consistent rate. However, as previously mentioned, there is no one solution which can achieve CR and  $\sigma$  minimal at the same time. PSOMOF algorithm is proposed by constructing the nondominated solutions which depicts the relation between  $\sigma$  and CR. Algorithm 1 shows the outline of PSOMOF algorithm. In this method, there are three steps in which each step uses PSO to get the result matrices:

- (1) *Minimize  $\sigma$  Step.* Firstly, each particle (there are 200 particles) generates its position and its velocity randomly. The position particle means that the particle generates randomly the candidate for the modified matrix. The element matrix can be represented only by the lower triangular matrix elements consecutively. The velocity particle means that the particle generates the value as adding/diminishing the position of the particles. The initial position of each particle  $Xi(t)$  is set the same as the original position. The initial velocity of each particle  $Vi(t)$  is set randomly but lower than 0.1. The best historical particle is defined as  $Xp$ , and the best position for all particles is defined as  $Xg$ . Initially,  $Xp$  is taken from the first position particle generated, while  $Xg$  is taken from the best position from the first position of all particles generated. In the next iteration, based on the previous velocity information,  $Xp$ ,  $Xg$ , and some variables ( $w, C_1, C_2, R_1, R_2$ ), the velocity of each particle is updated as described in (12). To set the value of variables, some experiments are conducted and the new position will be obtained based on the updated velocity, as described in (11). The evaluation of the fitness function is minimizing. However, if a particles  $\sigma$  is worse than before, or  $CR > 0.1$ , the update will be cancelled. The result of this fitness function also updates the new best historical position of each particle ( $Xp$ ) and the new best position of all particles as a group ( $Xg$ ). This process is repeated until the iteration maximum is reached.
- (2) *Minimize CR Step.* It is almost the same as step (1). If step (1) minimizes  $\sigma$  as its fitness function, then step (2) minimizes CR as its fitness function.
- (3) *Obtain a Set of Nondominated CR- $\sigma$  Solutions Step.* It is also the same as the process to minimize  $\sigma$ . Yet the process adds some various CRs, which are decreased gradually until CRmin is reached.

**3.2. Encoding and Fractioning of Original Element Matrix.** The encoding of matrix can be assembled by picking all

```

Initialize ()
Minimize  $\sigma$  ()
  For each particle generates the position and velocity randomly.
   $Xp \leftarrow$  the initial position //  $Xp$  is the particles best historical
   $Xg \leftarrow$  the initial position //  $Xg$  is the best of all particles
  Repeat
    Determine velocity using (12).
    Update new position particle using (11).
    Determine  $\sigma$  of new position using (10). If the new position has a lower  $\sigma$  and  $CR < 0.1$  updated new position is allowed
    otherwise, update new position is canceled and keeping the current position.
    Choosing the new  $Xp$  and  $Xg$  based on value  $\sigma$ 
  Until max iterations is reached
  Get minimal  $\sigma$ 
Minimize CR ()
  For each particle generates the position and velocity randomly.
   $Xp \leftarrow$  the initial position //  $Xp$  is the particles best historical
   $Xg \leftarrow$  the initial position //  $Xg$  is the best of all particles
  Repeat
    Determine velocity using (12).
    Update new position particle using (11).
    Determine CR of new position using (4). If the new position has a lower CR and  $CR < 0.1$  updated new position is allowed
    otherwise, update new position is canceled and keeping the current position.
    Choosing the new  $Xp$  and  $Xg$  based on value CR
  Until max iterations is reached
  Get minimal CR
Minimize CR- $\sigma$  ()
  CRo  $\leftarrow$  0.1
   $Xp \leftarrow$  the initial position //  $Xp$  is the particles best historical
   $Xg \leftarrow$  the initial position //  $Xg$  is the best of all particles
  While CRmin < CRo
    Repeat
      Determine velocity using (12).
      Update new position particle using (11).
      Determine CR of new position using (4). If the new position has a lower CR and  $CR < 0.1$  updated new position is allowed
      otherwise, update new position is canceled and keeping the current position.
      Choosing the new  $Xp$  and  $Xg$  based on value CR
    Until max iterations is reached
    Store the modified matrix and its CR,  $\sigma$ 
    CRo  $\leftarrow$  CRo -  $k$  //  $k$  is small value, in this study  $k = 0.001$ 
  End While
  Get matrices with their CR,  $\sigma$ 

```

ALGORITHM 1: PSOMOF algorithm.

elements in matrix. However, because the elements of fuzzy preference matrix (FPM) have a relation such that  $a_{ij} + a_{ji} = 1$  and  $a_{ii} = 0.5$ , encoding node can only encode the lower triangular elements of the matrix as nodes:

$$A = \begin{pmatrix} 0.5 & 0.6 & 0.4 & 0.8 \\ 0.4 & 0.5 & 0.3 & 0.7 \\ 0.6 & 0.7 & 0.5 & 0.9 \\ 0.2 & 0.3 & 0.1 & 0.5 \end{pmatrix} \quad (13)$$

Encode  $A = 0.4 - 0.6 - 0.7 - 0.2 - 0.3 - 0.1$ .

Equation (13) shows matrix  $A$  with  $n = 4$  and its encoding for FPM by picking row by row sequentially in the elements of the lower triangular matrix. The number element of encoding

$A$  can be determined  $(n^2 - n)/2$ . To obtain consistent matrix, of course, the value of each element matrix should be changed to a new value. The new values are chosen from the values of several candidates. Candidates elements are generated using the original fractioned value. If the original element is more than 0.5, the candidates will be between 0.5 and 1; if the original element is less than 0.5, the candidates will be between 0 and 0.5; if the original element is 0.5 (neutral), the candidate is still 0.5, or the original data should not be fractioned. This approach makes the candidate element not change the judgment tendency but will only change the judgment weight. The number of candidate elements is based on the fraction factor ( $\psi$ ). For example, if  $\psi = 0.01$ , then the number candidate will be  $50 (= (1 - 0.5)/0.01)$ . Suppose that matrix  $A$ ,  $a_r$  is one of the original elements on node  $r$ , and  $n$

TABLE 2: The original element and its candidate with  $\psi = 0.01$ .

Origin element	Candidate element
0.5	0.5
0, 0.1, 0.2, 0.3, 0.4	0, 0.01, 0.02, ..., 0.48, 0.49
0.6, 0.7, 0.8, 0.9, 1	0.51, 0.52, 0.53, ..., 0.99, 1

is the matrix size; thus, the sequence of nodes traveled,  $G_A$ , can be defined as

$$G_A = a_1, a_2, a_3, \dots, a_{(n^2-n)/2}. \quad (14)$$

Each element origin,  $a_r$ , is fractioned into several candidate elements  $a_{rs}$ , where  $s$  denotes the index of the candidate element, as described in

$$a_{rs} = a_{rs-1} + \psi$$

$$a_{r0} = \begin{cases} 0, & \text{if } 0 \leq a_r < 0.5, \\ 0.5, & \text{if } 0.5 < a_r \leq 1, \end{cases} \quad (15)$$

where  $r = 1, 2, 3, \dots, (n^2 - n)/2$ ;  $s = 1, 2, 3, \dots, (1 - 0.5)/\psi$ .

Table 2 shows the original element and its candidate as a result of being fractioned if  $\psi = 0.01$ . There are 50 candidates to substitute for the origin element.

These fractioned elements can be used as candidate nodes to travel by particle in PSOMOF. The particle will move from the candidate in one node to the candidate in the next node. However, it is possible that the particle preserves the original element.

**3.3. Determining CI in a Fuzzy Preference Matrix.** Determining CI on fuzzy preference matrix by using (8) as a transforming from (7) is not suitable. Transformation using operation  $b_{ij} = a_{ij}/a_{ji}$  is not appropriate because it can exceed the threshold of the multiplicative matrix element. For example, suppose  $a_{ij} = 0.95$  and  $a_{ji} = 0.05$ . By transforming the above formula,  $b_{ij}$  will be 19. The value  $b_{ij}$  exceeds 9, which is the threshold of the multiplicative element matrix. Therefore, in this study, we use a method to transform the fuzzy preference ( $a_{ij}$ ) to the multiplicative preference ( $b_{ij}$ ) as introduced by Herrera-Viedma et al. [17] to determine the multiplicative consistency as shown in

$$b_{ij} = 9^{2 \cdot a_{ij}}, \quad (16)$$

$$a_{ij} = \frac{1}{2} \left( 1 + \log_9 b_{ij} \right). \quad (17)$$

Therefore, if the element fuzzy preference matrix  $a_{ij} = 0.95$ , it can be transformed to be the element multiplicative matrix  $b_{ij} = 7.22$ . This transformation value is not higher than the maximum scale of 9. By using (16), a new formula is proposed to determine the CI, as shown in

$$CI = \frac{1}{n(n-1)} \sum_{1 \leq i < j \leq n} \left[ 9^{2 \cdot a_{ij}-1} \cdot \frac{w_j}{w_i} + 9^{2 \cdot a_{ji}-1} \cdot \frac{w_i}{w_j} - 2 \right]. \quad (18)$$

TABLE 3: Parameter settings for the PSOMOF, NSGA-2, and MOPSO.

Parameter	Value
PSOMOF:	
$W$	0.1
$C_1$	0.2
$C_2$	0.3
$R_1$	0.4
$R_2$	0.5
NSGA-2	
Population size	100
Generation	200
Rate crossover	0.9
Rate mutation	0.1
MOPSO	
Number of particle	20
Number of cycles	1000

To prove this formula, the consistent ratio rate of one sample matrix as shown in (13) is determined. This sample matrix is selected from Xu et al. [6]. According to (5), obviously, matrix  $A$  can be verified as a consistent matrix. Contrarily, matrix  $A$  is identified as an inconsistent matrix with  $CR = 0.11$  when it is determined by (4) and (7). However, if (4) and (18) are used,  $CR$  will be 0.05 and therefore will be a consistent matrix as in the result of (5). Therefore, in this research, (18) is used to define the CI value.

## 4. Experimental Results

**4.1. Parameter Setting.** Table 3 shows the parameter settings for the proposed and compared methods. The inconsistent matrix can be taken from the real life application which needs the decision maker opinions of comparing several criteria to get some alternatives. Once the matrix is identified as inconsistent, PSOMOF is able to be used to repair the inconsistent matrix. To see the performance of proposed methods in repairing inconsistent matrices, there are 15 inconsistent fuzzy preference matrices which need to be repaired as shown in Table 4. Some matrices come from the other papers, but some matrices are created randomly.

**4.2. Generating Nondominated Solutions.** As aforementioned, there are two objectives for this proposed method, that is, the best  $CR$  and deviation matrix. Both objectives will conflict each other. When the  $CR$  is lowest (good consistent ratio), it leads to the highest (the worst) deviation, and vice versa. However, in order to get the acceptable matrix, the  $CR$  of modified matrix is limited below 0.1. It makes the solution consist of some relations ("CR-deviation") which can be identified as nondominated solutions. Equations (19a), (19b), and (19c) display the performance of PSOMOF to get the best  $CR$ - $\sigma$  for  $A_5$ , respectively. The origin matrix  $A_5$  (19a) can be transformed to the modified matrices which have the best  $CR$  (19b) and  $\sigma$  (19c), respectively:

TABLE 4: The dataset inconsistency matrices.

Matrix	Elements of lower triangular matrix	CR
	Size $4 \times 4$	
$A_1$	0.9-0.4-0.2-0.3-0.6-0.1 <sup>a</sup>	0.687
$A_2$	0.8-0.4-0.1-0.1-0.3-0.7	0.364
$A_3$	0.4-0.6-0.4-0.7-0.4-0.3 <sup>b</sup>	0.183
$A_4$	0.4-0.3-0.4-0.3-0.1-0.9	0.427
	Size $5 \times 5$	
$A_5$	0.4-0.3-0.4-0.7-0.8-0.2-0.4-0.4-0.6-0.2	0.319
$A_6$	0.1-0.2-0.3-0.9-0.6-0.8-0.7-0.4-0.6-0.3	0.343
$A_7$	0.7-0.2-0.1-0.3-0.8-0.8-0.7-0.1-0.6-0.4	0.359
$A_8$	0.1-0.3-0.1-0.8-0.8-0.4-0.6-0.8-0.6-0.7	0.479
	Size $6 \times 6$	
$A_9$	0.8-0.2-0.1-0.4-0.8-0.9-0.4-0.2-0.4-0.7-0.9-0.8-0.7-0.4-0.3	0.440
$A_{10}$	0.3-0.1-0.8-0.8-0.3-0.7-0.2-0.4-0.4-0.7-0.7-0.6-0.4-0.8-0.1	0.531
$A_{11}$	0.2-0.8-0.1-0.7-0.8-0.4-0.4-0.6-0.7-0.6-0.1-0.4-0.6-0.3-0.7	0.437
	Size $7 \times 7$	
$A_{12}$	0.7-0.2-0.4-0.7-0.3-0.6-0.4-0.3-0.9-0.2-0.7-0.4-0.6-0.8-0.8-0.8-0.3-0.9-0.2-0.7-0.9	0.315
$A_{13}$	0.7-0.8-0.3-0.4-0.6-0.7-0.2-0.7-0.2-0.3-0.8-0.3-0.3-0.2-0.6-0.4-0.7-0.3-0.2-0.1-0.7	0.353
	Size $8 \times 8$	
$A_{14}$	0.8-0.8-0.8-0.3-0.6-0.8-0.7-0.7-0.4-0.7-0.7-0.9-0.7-0.4-0.4-0.4-0.3-0.3-0.4-0.7-0.2-0.6-0.2-0.2-0.8-0.7-0.2-0.7	0.313
$A_{15}$	0.7-0.8-0.7-0.3-0.8-0.6-0.4-0.7-0.2-0.2-0.7-0.3-0.8-0.7-0.3-0.1-0.3-0.1-0.2-0.8-0.8-0.3-0.8-0.1-0.6-0.2-0.1-0.4	0.457

Data on a and b is picked from [8, 18].

CR = 0.319

$$\begin{pmatrix} 0.5 & 0.6 & 0.7 & 0.3 & 0.6 \\ 0.4 & 0.5 & 0.6 & 0.2 & 0.6 \\ 0.3 & 0.4 & 0.5 & 0.8 & 0.4 \\ 0.7 & 0.8 & 0.2 & 0.5 & 0.8 \\ 0.4 & 0.4 & 0.6 & 0.2 & 0.5 \end{pmatrix}. \quad (19a)$$

CR = 0.003 and  $\sigma = 0.161$

$$\begin{pmatrix} 0.5 & 0.5146 & 0.5072 & 0.4923 & 0.5190 \\ 0.4854 & 0.5 & 0.5088 & 0.4504 & 0.5097 \\ 0.4928 & 0.4912 & 0.5 & 0.5177 & 0.4882 \\ 0.5077 & 0.5496 & 0.4823 & 0.5 & 0.5312 \\ 0.4810 & 0.4903 & 0.5112 & 0.4688 & 0.5 \end{pmatrix}. \quad (19b)$$

CR = 0.099 and  $\sigma = 0.073$

$$\begin{pmatrix} 0.5 & 0.5717 & 0.6188 & 0.3855 & 0.6477 \\ 0.4283 & 0.5 & 0.52 & 0.26 & 0.5982 \\ 0.3812 & 0.4473 & 0.5 & 0.6630 & 0.4645 \\ 0.6145 & 0.6554 & 0.3370 & 0.5 & 0.7341 \\ 0.3523 & 0.4018 & 0.5355 & 0.2659 & 0.5 \end{pmatrix}. \quad (19c)$$

PSOMOF splits the method into three steps. These are to find the optimal deviation, optimal CR, and the optimal deviation with the particular value of CR. Figure 1 shows

the process convergence to find the optimal deviation, while Figure 2 shows process convergence to find the optimal CR. Both of them are conducted on  $A_5$ .

After obtaining the minimal CR and  $\sigma$ , the third step of the PSOMOF is executed to get the nondominated CR-deviation nodes. By using PSOMOF, for each CR, the optimal deviation can be obtained. This proposed method thus successfully generates some nodes as solutions. Figure 3 shows the Pareto graph which depicts the relation of CR and deviation of matrix. The sample matrices for fuzzy preference matrix are  $A_1, A_5, A_9, A_{12}$ , and  $A_{14}$ . It shows clearly that they will be contradictory to each other. In case of matrices, when  $\sigma$  is minimized, CR is maximized. Likewise, when CR is minimized,  $\sigma$  is maximized.

**4.3. Comparison with Other Methods.** To evaluate the performance of PSOMOF, this study uses the metric analysis [21, 22]. The performance is represented by the Pareto graph 10 times. The Pareto graph is then compared with Pareto graphs of two other algorithms, NSGA-2 [23] and MOPSO [24]. The Pareto-optimal set is generated by merging all of the Pareto graphs of all algorithms (PSOMOF, NSGA-2, and MOPSO) into a single Pareto solution. The nondominated solutions for each algorithm are generated by executing each algorithm once on a sample inconsistent matrix ( $A_1, A_5, A_9, A_{12}$ , and  $A_{14}$ ). There are 3 metrics to measure the performance of nondominated solutions achieved using the proposed method. Suppose a set of nondominated solutions  $X \subseteq X$ .

$M_1$  Metric. This metric measures the average distance of the resulting nondominated set solutions to the Pareto-optimal

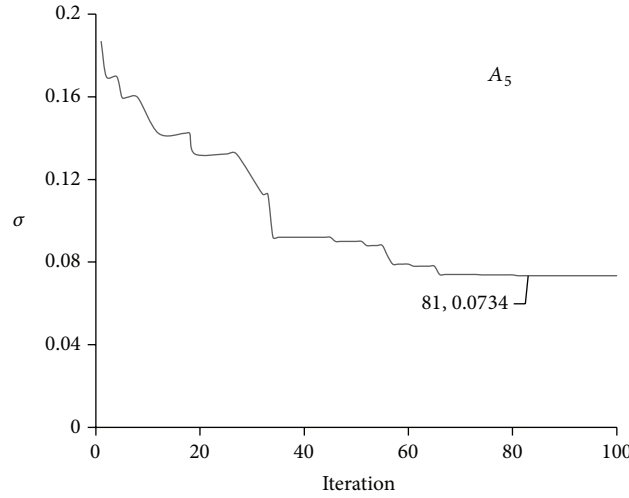


FIGURE 1: The process convergence to find the optimal deviation on  $A_5$ .

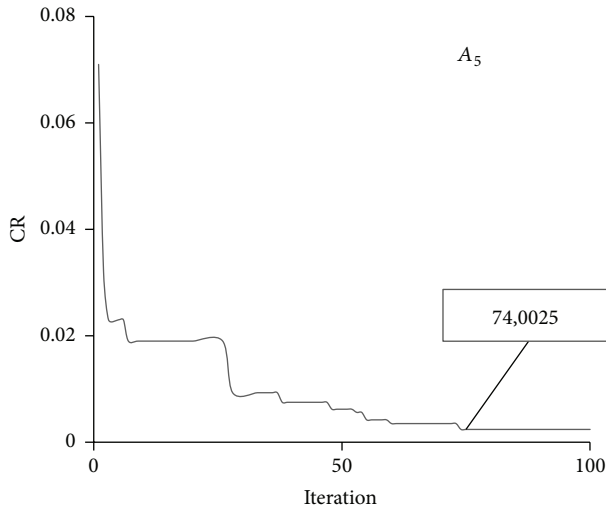


FIGURE 2: The process convergence to find the optimal CR on  $A_5$ .

set solutions. The better value should be a lower  $M_1$ . It can be defined as described in

$$M_1(X') = \frac{1}{|X'|} \sum_{a' \in x'} \min \{ \|a' - \bar{a}\| \mid a \in \bar{X} \}. \quad (20)$$

$M_2$  Metric. This metric measures the number of distribution nondominated solutions which are covered by a neighbourhood parameter  $d > 0.03$ . A bigger  $M_2$  indicates better performance.  $M_2$  can be defined as

$$M_2(X') = \frac{1}{|X' - 1|} \sum_{a' \in x'} |\{b' \in x'; \|a' - b'\| > d\}| \quad (21)$$

$M_3$  Metric. This metric measures the extent of nondominated sets obtained. A wide range of values should be covered by the

nondominated solutions. The bigger  $M_3$  is better.  $M_3$  can be defined as

$$M_3(X') = \sqrt{\sum_{i=1}^m \max \{ \|a'_i - b'_i\|; a', b' \in X' \}}. \quad (22)$$

The comparison results are shown in Table 5. It shows that PSOMOF  $M_1$  metric is minimal in all of matrices compared to MOPSO and NSGA-2. These results show that most of the Pareto graphs of PSOMOF are closer to the Pareto-optimal front than both algorithms (NSGA-2 and MOPSO). For  $M_2$  metric, the PSOMOF result is larger than both of the other algorithms except for  $A_{12}$ . This indicates that the solutions of the proposed method are more distributed than both algorithms. In the  $M_3$  metric, the proposed algorithm also outperforms as compared to the NSGA-2 and MOPSO. The proposed method returned nondominated solutions further than both of the other algorithms. Regarding this result, the proposed method, PSOMOF, can be claimed as the better algorithm compared to the two algorithms (NSGA-2 and MOPSO).

## 5. Conclusions

This paper presents a study to use the multiobjective PSO to solve the inconsistent fuzzy preference matrix in AHP, called PSOMOF. There are two objectives (consistent ratio and deviation matrix) considered in rectifying the matrix in order to be consistent. However, they are conflicting in that process. Therefore, the proposed algorithms offer some nondominated solutions which also satisfied the acceptable consistent matrices. The process in PSOMOF is split into three parts in which each part applies the PSO process. To see the performance, 15 inconsistent comparison matrices are repaired by the proposed methods. Besides repairing inconsistent comparison matrices, the proposed method also can

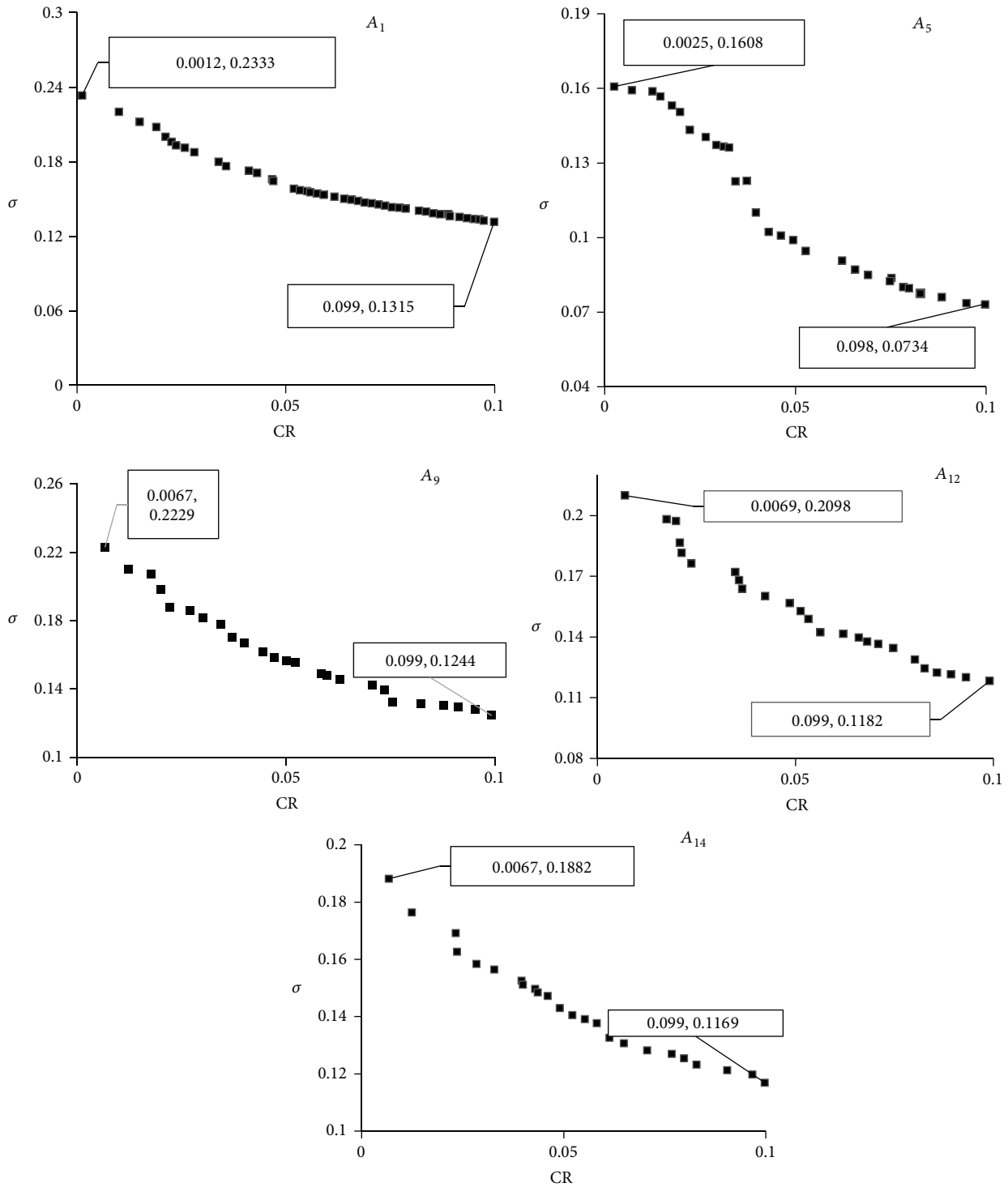


FIGURE 3: The Pareto graph solutions which show relation CR- $\sigma$ .

generated some nondominated solution which can be classified as optimal solutions. This result shows the PSO algorithm is the potential approach to solve the inconsistent comparison matrix in AHP. The other intelligent algorithm also might be used to solve this problem. Further, this proposed method might be a potential method to combine with other method metaheuristic (hybrid) to improve the quality of results.

**Conflict of Interests**

The authors declare that there is no conflict of interests regarding the publication of this paper.

**Acknowledgments**

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions on the paper.



TABLE 5: Comparison of the metric performance of NSGA-2, MOPSO, and MOBAF.

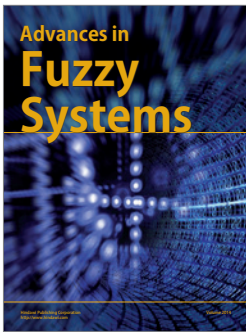
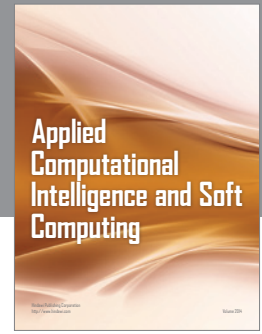
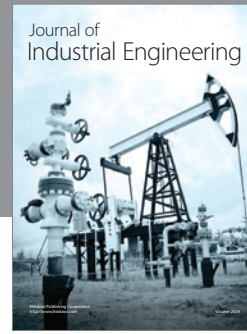
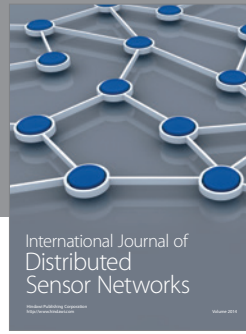
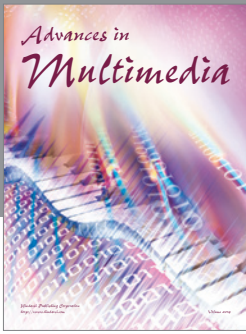
Method	$A_1$	$A_5$	$A_9$	$A_{12}$	$A_{14}$
NSGA-2					
$M_1$	0.000896	0.000913	0.00127	0.00146	0.00133
$M_2$	24.0	26.2	15.9	20.4	14.1
$M_3$	1.76	1.50	1.34	1.21	1.11
MOPSO					
$M_1$	0.000830	0.000701	0.00110	0.000930	0.00122
$M_2$	28.7	<b>27.9</b>	19.6	<b>26.1</b>	17.7
$M_3$	1.89	1.64	1.60	1.39	1.29
PSOMOF					
$M_1$	<b>0.000728</b>	<b>0.000688</b>	<b>0.000957</b>	<b>0.000926</b>	<b>0.000998</b>
$M_2$	<b>32.5</b>	<b>27.9</b>	<b>20.6</b>	25.7	<b>19.0</b>
$M_3$	<b>1.98</b>	<b>1.92</b>	<b>1.65</b>	<b>1.58</b>	<b>1.48</b>

This work was supported in part by the Ministry of Science and Technology of Taiwan, under Contracts MOST103-2221-E-197-034, MOST 104-2221-E-197-005, and NSC 100-2218-E-006-028-MY3.

## References

- [1] T. L. Saaty, *The Analytic Hierarchy Process: Planning, Priority Setting, Resources Allocation*, McGraw-Hill, New York, NY, USA, 1980.
- [2] S. A. Orlovsky, "Decision-making with a fuzzy preference relation," *Fuzzy Sets and Systems*, vol. 1, no. 3, pp. 155–167, 1978.
- [3] F. Chiclana, F. Herrera, and E. Herrera-Viedma, "Integrating multiplicative preference relations in a multipurpose decision-making model based on fuzzy preference relations," *Fuzzy Sets and Systems*, vol. 122, no. 2, pp. 277–291, 2001.
- [4] Y. Xu and H. Wang, "Eigenvector method, consistency test and inconsistency repairing for an incomplete fuzzy preference relation," *Applied Mathematical Modelling*, vol. 37, no. 7, pp. 5171–5183, 2013.
- [5] Z. Xu and J. Chen, "Group decision-making procedure based on incomplete reciprocal relations," *Soft Computing*, vol. 12, no. 6, pp. 515–521, 2008.
- [6] Y. Xu, Q. Da, and H. Wang, "A note on group decision-making procedure based on incomplete reciprocal relations," *Soft Computing*, vol. 15, no. 7, pp. 1289–1300, 2011.
- [7] Y. Xu, J. N. D. Gupta, and H. Wang, "The ordinal consistency of an incomplete reciprocal preference relation," *Fuzzy Sets and Systems*, vol. 246, pp. 62–77, 2014.
- [8] X. Liu, Y. Pan, Y. Xu, and S. Yu, "Least square completion and inconsistency repair methods for additively consistent fuzzy preference relations," *Fuzzy Sets and Systems*, vol. 198, pp. 1–19, 2012.
- [9] S.-M. Chen, T.-E. Lin, and L.-W. Lee, "Group decision making using incomplete fuzzy preference relations based on the additive consistency and the order consistency," *Information Sciences*, vol. 259, pp. 1–15, 2014.
- [10] F. Chiclana, E. Herrera-Viedma, F. Alonso, and S. Herrera, "Cardinal consistency of reciprocal preference relations: a characterization of multiplicative transitivity," *IEEE Transactions on Fuzzy Systems*, vol. 17, no. 1, pp. 14–23, 2009.
- [11] M. Xia, Z. Xu, and J. Chen, "Algorithms for improving consistency or consensus of reciprocal [0,1]-valued preference relations," *Fuzzy Sets and Systems*, vol. 216, pp. 108–133, 2013.
- [12] I.-T. Yang, W.-C. Wang, and T.-I. Yang, "Automatic repair of inconsistent pairwise weighting matrices in analytic hierarchy process," *Automation in Construction*, vol. 22, pp. 290–297, 2012.
- [13] C.-C. Lin, W.-C. Wang, and W.-D. Yu, "Improving AHP for construction with an adaptive AHP approach ( $A^3$ )," *Automation in Construction*, vol. 17, no. 2, pp. 180–187, 2008.
- [14] A. S. Girsang, C.-W. Tsai, and C.-S. Yang, "Ant algorithm for modifying an inconsistent pairwise weighting matrix in an analytic hierarchy process," *Neural Computing and Applications*, vol. 26, no. 2, pp. 313–327, 2014.
- [15] A. S. Girsang, C.-W. Tsai, and C.-S. Yang, "Ant colony optimization for reducing the consistency ratio in comparison matrix," in *Proceedings of the International Conference on Advances in Engineering and Technology (ICAET '14)*, pp. 577–582, Singapore, March 2014.
- [16] A. S. Girsang, C. Tsai, and C. Yang, "Multi-objective particle swarm optimization for repairing inconsistent comparison matrices," *International Journal of Computers and Applications*, vol. 36, no. 3, 2014.
- [17] E. Herrera-Viedma, F. Herrera, F. Chiclana, and M. Luque, "Some issues on consistency of fuzzy preference relations," *European Journal of Operational Research*, vol. 154, no. 1, pp. 98–109, 2004.
- [18] Z. Xu and Q. Da, "An approach to improving consistency of fuzzy preference matrix," *Fuzzy Optimization and Decision Making*, vol. 2, no. 1, pp. 3–12, 2003.
- [19] Z. Xu, "On consistency of the weighted geometric mean complex judgement matrix in AHP," *European Journal of Operational Research*, vol. 126, no. 3, pp. 683–687, 2000.
- [20] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks*, vol. 4, pp. 1942–1948, December 1995.
- [21] C. García-Martínez, O. Cordón, and F. Herrera, "A taxonomy and an empirical analysis of multiple objective ant colony optimization algorithms for the bi-criteria TSP," *European Journal of Operational Research*, vol. 180, no. 1, pp. 116–148, 2007.
- [22] E. Zitzler, K. Deb, and L. Thiele, "Comparison of multiobjective evolutionary algorithms: empirical results," *Evolutionary Computation*, vol. 8, no. 2, pp. 173–195, 2000.

- [23] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.
- [24] C. A. Coello Coello and M. S. Lechuga, "MOPSO: a proposal for multiple objective particle swarm optimization," in *Proceedings of the Congress on Evolutionary Computation (CEC '02)*, vol. 2, pp. 1051–1056, Honolulu, Hawaii, USA, May 2002.



**Hindawi**

Submit your manuscripts at  
<http://www.hindawi.com>

