Immune Evolutionary Algorithm for Determining Weights in Interval Judgments

Jingwen Huang[2,3] **Yongsheng Ding**[1,2] **Zhihua Hu**[1]

[1]College of Information Sciences and Technology, Donghua University, Shanghai 201620, P. R. China
[2]Glorious Sun School of Business and Management, Donghua University, Shanghai 200051, P. R. China
[3]Information Network Center, Guangxi University, Nanning 530004, P. R. China

Abstract

In a Multiple Criteria Decision Analysis (MCDA) problem, determination of weights is an important aspect of the Analytic Hierarchy Process (AHP). In this paper, the problem to computer the weight vector of the interval number matrix is regarded as a nonlinear optimization model with constraints. An immune evolutionary algorithm is proposed to solve the problem with high speed by the matrix based coding technique and several immune operators including clonal proliferation, hypermutation, clonal elimination and immune memory. The experiments show the algorithm has the promising performance of convergence and accuracy.

Keywords: Immune evolutionary algorithm, Interval judgments, Weights

1. Introduction

Decision making is a typical intellectual activity, which is inherently a black box but can be viewed as a multi-criteria satisfaction process. As we all well know, the criteria often conflict with each other. So, it is very difficult to satisfy all the criteria simultaneously. Over the Decades, there have been lots of attempts at devoting to the solving methodologies.

In the late 1970s, Saaty introduced the Analytic Hierarchy Process (AHP) as a multi-criteria decision support tool [1]. Subsequently, the AHP has been widely used in a variety of fields because of its conceptual simplicity and its capability of handling subjective criteria and inconsistencies in the decision making process.

According to Saaty [1], AHP is based on comparing n objects(alternatives or criteria) in pairs derive the underlying weights from comparison matrices by using a suitable weight determination technique. In order to reflect expert's judgments under the conditions of uncertainty, lots of works have improved the AHP by allowing interval numbers for the pairwise comparisons. The present paper is concerned with this topic. We will propose an immune evolutionary algorithm for determining weights in interval judgment matrices.

The remainder of the paper is structured as follows. Section 2 describes the previous works and problems. An immune evolutionary algorithm for determining weights in interval judgment matrices is put forth in Section 3. In Section 4, we apply the proposed method to a 4×4 interval judgments matrix and discuss the results. Finally, in Section 5, we present our conclusions.

2. Previous works and limitations

The Saaty's classical AHP uses crisp values to represent the decision maker's judgments, and the priority weights are obtained by the eigenvector method from a pairwise comparison matrix of judgments [1]. However, this is not appropriate in many practical real problems under the conditions of uncertainty. Due to the complexity and uncertainty involved in real world decision making, it is sometimes unrealistic or impossible to acquire exact judgments. It is more natural or easier to provide interval or fuzzy judgments for parts or all of the judgments in a pairwise comparison matrix.

With interval judgments, a decision maker uses approximate ratio statements (as intervals of values on a ratio scale) and makes 'strength of preference' statements consistent with 'natural verbal expressions' such as "the *i*th subattribute is two to three times more important than the *j*th subattribute". Interval judgments have been widely used in AHP to express subjective uncertainty in preference.

It is easy to give interval comparisons since they are suitable to represent uncertain human judgments and based on human intuition. However, there exists a serious problem that pairwise comparisons might be inconsistent with each other [2] so that we can not obtain weights by eigenvector method ordinarily.

AHP with interval judgments has been studied by Arbel [3], Saaty and Vargas [2], Islam [4] et al. Arbel viewed interval judgments as linear constraints on weights and formulated the weight estimation problem as a linear programming (LP) model, which is not always feasible for inconsistent interval comparison matrices [5, 6]. It is complex because Arbel's approach is based on solving problems on all vertices for obtaining interval weights.

From Arbel's work [3], a number of techniques have been developed to deal with interval comparison matrices. Recently, researchers focus on the methods based on goal programming (GP) [7], including lexicographic goal programming (LGP) method [8], and two-stage logarithmic goal programming (TLGP) method [9, 10], et al. It was reported that the computation of the methods based on GP is much fast if the criteria are not more than five [11] but there is a obvious shortcoming lies in that there are too many constraints in the GP model to put into practice.

3. The proposed algorithm

In this section, we propose An immune evolutionary algorithm for determining weights in interval judgment matrices. Firstly, we review the eigenvector method for obtaining crisp weights. Based on these, we describe interval judgment matrix referring to Saaty's 1-9 points ratio scale, and research a nonlinear optimization model to obtain the weights vector of the interval matrix. Thirdly, we describe the steps of the proposed algorithm and some problems relating to it.

3.1. Crisp weights by eigenvector method

In Saaty's AHP, when there are n objects(alternatives or criteria), a decision maker compares a pair of objects for all possible pairs to obtain a positive reciprocal pairwise comparison matrix A as follows:

$$A=(a_{ij})=\begin{pmatrix} 1 & a_{12} & \cdots & a_{1n} \\ a_{21} & 1 & \cdots & a_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ a_{n1} & a_{n2} & \cdots & 1 \end{pmatrix}$$

where a_{ij} shows the priority ratio of object i comparing to object j and they satisfy the following relations.

(I) $a_{ii}=1,\forall i$

(II) $a_{ij}=\dfrac{1}{a_{ji}},\forall i,j,i\neq j$

When the pairwise comparison matrix A satisfies the consistency measure, called the consistency ratio, given by

$$CR(A)=\frac{\lambda-n}{(n-1)RI} \tag{1}$$

where RI is the random consistency index for A and λ is the maximum eigenvalue, the weights vector w is obtained by eigenvector method formulated as follows.

$$Aw=\lambda w \tag{2}$$

where λ is the maximum eigenvalue and w is the eigenvector corresponding to λ . Noted that w is normalized to be one.

For the measure CR, Saaty proposed the 0.10 threshold (0.05 for n = 3 and 0.08 for n = 4) to accept the estimation of the weight vector w. When the CR is greater than 0.10, inconsistent judgments must be modified in order to improve the consistency.

3.2. Optimization model

In this paper, we present judgments by allowing interval numbers for the pairwise comparisons as follows:

$$A^I=(a^I_{ij})=\begin{pmatrix} [1,1] & [\underline{a}_{12},\overline{a}_{12}] & \cdots & [\underline{a}_{1n},\overline{a}_{1n}] \\ [\underline{a}_{21},\overline{a}_{21}] & [1,1] & \cdots & [\underline{a}_{2n},\overline{a}_{2n}] \\ \cdots & \cdots & \cdots & \cdots \\ [\underline{a}_{n1},\overline{a}_{n1}] & [\underline{a}_{n2},\overline{a}_{n2}] & \cdots & [1,1] \end{pmatrix}$$

where $\underline{a}_{ij}$ and $\overline{a}_{ij}$ presents the lower bound and upper bound of a^I_{ij}, respectively.

According to Saaty's 1-9 points ratio scale [1] and the rules of interval numbers [3], the above interval comparison matrix should satisfy the following relations.

(I) $a^I_{ii}=1,i=1,2...,n$

(II) $a^I_{ij}=\left[\underline{a}_{ij},\overline{a}_{ij}\right],\dfrac{1}{9}\le\underline{a}_{ij}\le\overline{a}_{ij}\le 9,\forall i,j$

(III) $a_{ij}^{I} = \frac{1}{a_{ij}^{I}} = [\frac{1}{\overline{a_{ji}}}, \frac{1}{\underline{a_{ji}}}], \forall i, j$

In order to generate interval weights by eigenvector method, we split the above interval comparison matrix into two crisp nonnegative matrices as $A^I = [\underline{A}, \overline{A}]$, where

$$\underline{A} = (\underline{a_{ij}}) = \begin{pmatrix} 1 & \underline{a_{12}} & \cdots & \underline{a_{1n}} \\ \underline{a_{21}} & 1 & \cdots & \underline{a_{2n}} \\ \cdots & \cdots & \cdots & \cdots \\ \underline{a_{n1}} & \underline{a_{n2}} & \cdots & 1 \end{pmatrix}$$

and

$$\overline{A} = (\overline{a_{ij}}) = \begin{pmatrix} 1 & \overline{a_{12}} & \cdots & \overline{a_{1n}} \\ \overline{a_{21}} & 1 & \cdots & \overline{a_{2n}} \\ \cdots & \cdots & \cdots & \cdots \\ \overline{a_{n1}} & \overline{a_{n2}} & \cdots & 1 \end{pmatrix}$$

Let $\lambda = [\underline{\lambda}, \overline{\lambda}]$ be the principal eigenvalue of A^I , which is an interval number. The principal right eigenvector corresponding to λ is denoted by

$$w = [\underline{w}, \overline{w}] = ([\underline{w_1}, \overline{w_1}], ..., [\underline{w_n}, \overline{w_n}])^T \qquad (3)$$

where

$\underline{w} = (\underline{w_1}, ..., \underline{w_n})^T$ and $\overline{w} = (\overline{w_1}, ..., \overline{w_n})^T$.

According to Sugihara [12], an interval weight vector as EQ. (3)is said to be normalized if and only if

$$\sum_i \overline{w_i} - \max_j (\overline{w_j} - \underline{w_j}) \geq 1 \qquad (4)$$

$$\sum_i \underline{w_i} + \max_j (\overline{w_j} - \underline{w_j}) \leq 1 \qquad (5)$$

The meaning of EQ. (4) and EQ. (5) is to make each weight interval as wide as possible. Thus, we propose a model for determining interval weights as follows.

Denote $M = \{1, 2, ..., m\}$. For the interval judgment matrix A^I , m judgment matrix are generated, and the maximum eigenvalue as well as the corresponding eigenvector for each generated matrix is computed. Then, the eigenvalue is standardized to $w^i, i \in M$. For each component w_j^i of w^i, $j \in N$, the low and upper bounds can be denoted and computed as: $\underline{w_j} = \min\left\{ w_j^i \middle| i \in M \right\}$, $\overline{w_j} = \max\left\{ w_j^i \middle| i \in M \right\}$. Then, the interval weight vector of A^I is given by $w^I = \left(w_j^I \right)_{n \times 1} = \left(\left[\underline{w_j}, \overline{w_j} \right] \right)_{n \times 1}$. The components of w^I is determined by weight vectors of all $A^i, i \in M$, then the problem to computer the interval weight vector is transferred to an constrained optimization problem with the form of (6) and (7).

P1 $\underline{w_j} = \min\left\{ w_j^i \middle| i \in M \right\}$

s.t.

$$\begin{aligned} & A^i w^i = \lambda^i w^i \\ & CR\left(A^i\right) \leq 0.1 \\ & i \in M \end{aligned} \qquad (6)$$

P2 $\overline{w_j} = \max\left\{ w_j^i \middle| i \in M \right\}$

s.t.

$$\begin{aligned} & A^i w^i = \lambda^i w^i \\ & CR\left(A^i\right) \leq 0.1 \\ & i \in M \end{aligned} \qquad (7)$$

It is obvious to solve the problem defined by (6) and (7). In order to computer the weight vector quickly, it is vital to construct the sequence of $A^i, i \in M$, in other words, the travel of the space of the deterministic judgment matrix by the intervals. And the convergence to the bounds of the weight vector should be as quick as possible. In this paper, an immune evolutionary algorithm is designed to solve the problem by the global convergence ability of immune system [13]-[15]. In the immune system, antibodies detect the antigen and help the immune cells to destroy it. After the detection of antigen, the antibody will be selected to be proliferated to generate a large number of antibodies. And the generated antibodies will endure hypermutation. After the antigen is destroyed, the antibodies with lower affinity will be eliminated to keep the antibody population stable. At the same time, the superiors of the antibodies will be kept in the immune memory.

Mapped to the optimization problem in this paper, the objective is the antigen, and the randomly determined judgment matrix is antibody. By various immune operators and immune memory the antigen is discovered and the interval weight vector is computed quickly.

3.3. Algorithm

The proposed immune evolutionary algorithm in this paper is consisted of the following steps:

Step1: Initialize the interval judgment matrix and various parameters;

Step2: Initialize the antibody population with the size of $\mathrm{m}_{\mathrm{pop}}$;

Step3: Select the superior antibodies to proliferate;

Step4: The proliferated antibodies endure mutation with higher probability (m_{hyperm});

Step5: The inferior antibodies with higher affinity are chosen to be eliminated to keep the antibody population stable;

Step6: The superior antibodies with higher affinity are chosen to be kept in the immune memory;

Step7: If the termination criteria are satisfied, the algorithm is stopped and the superior antibody is outputted as the antigen; or else goes to Step3

Several concepts and operators of the algorithm are as follows:

(1) The coding style of antibody. The matrix coding is used to keep the consistency with the judgment matrix. The component of the matrix is coded as real number. Such coding style helps to traverse the whole space determined by the interval judgment matrix.

(2) The affinity function. In this paper, the affinity is defined by the objective function. And the satisfactory consistency condition is adapted to eliminate the antibodies.

(3) Clonal proliferation. At first, the superior antibodies are chosen with the ratio m_{select}, then, the arithmetical crossover operator is performed on every two antibodies to generate m_{clonal} new antibodies. Arithmetical crossover on two real numbers a and b to generate c is defined as $c = \alpha a + (1-\alpha)b$, where α is generated by random between 0 and 1.

(4) Hypermutation. The hypermutation is performed on the proliferated antibodies with a high probability(m_{hyperm}). If the low and the upper bounds are L and U, the mutation is defined to generate c: $c = c = L + \alpha\left(U - L\right)$, where α is generated by random between 0 and 1.

(5) Clonal elimination. The inferior antibodies will be chosen to be eliminated to keep the population stable.

(6) Immune memory: in fact the immune memory is a elitism strategy to store the superior antibody.

(7) After $\mathrm{m}_{\mathrm{gen}}$ generations, the algorithm stops to generate the results.

4. Computational experiment

In this section, we apply the proposed algorithm to a 4 ×4 interval judgments matrix as follows, which was given by Arbel and Vargas [3].

$$A^I = \begin{pmatrix} [1,1] & [2,5] & [2,4] & [1,3] \\ [1/5,1/2] & [1,1] & [1,3] & [1,2] \\ [1/4,1/2] & [1/3,1] & [1,1] & [1/2,1] \\ [1/3,1] & [1/2,1] & [1,2] & [1,1] \end{pmatrix}$$

4.1. Results

In the proposed algorithm, we set the parameters as Table 1, and obtain weights vector as Table 2.

Parameter	Value
The size of antibody population(m_{pop})	40
The iterations(m_{gen})	100
The select ratio for clonal proliferation(m_{select})	0.25
The coefficientof clonal proliferation(m_{clonal})	2
Rate of hypermutation(m_{hyperm})	0.5

Table 1: Parameters setting of the proposed algorithm.

Our Results			Results from Arbel		
w	lower	upper	w	lower	upper
w_1	0.341	0.604	w_1	0.369	0.552
w_2	0.140	0.328	w_2	0.150	0.290
w_3	0.091	0.225	w_3	0.093	0.189
w_4	0.122	0.311	w_4	0.133	0.260

Table 2: Results.

From the comparison in Table 2, the results in this paper are better than the results in [3]. The distribution range of the weight vector are more wider to present the uncertainty.

4.2. Efficiency of the algorithm

Convergence profiles for determining weights vector by using the proposed algorithm are displayed in Fig. 1 to Fig. 8 as follows.

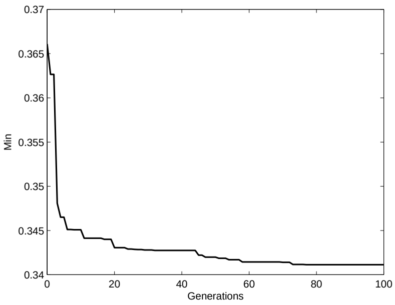

Fig. 1: Convergence profiles for determining lower bounds of w_1.

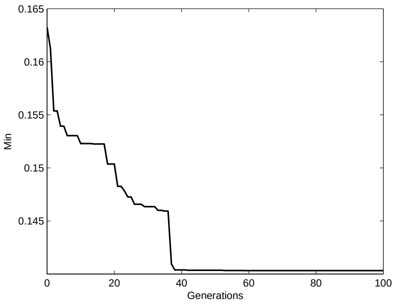

Fig. 2: Convergence profiles for determining lower bounds of w_2.

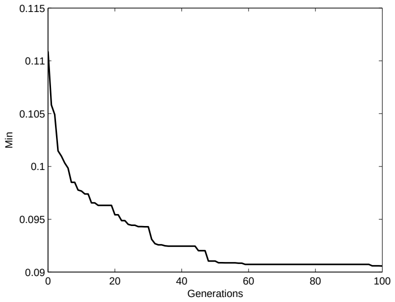

Fig. 3: Convergence profiles for determining lower bounds of w_3.

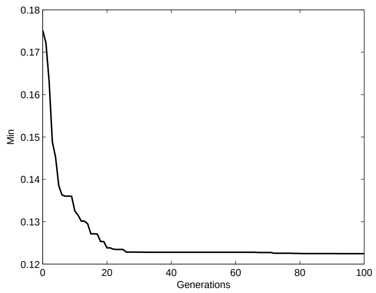

Fig. 4: Convergence profiles for determining lower bounds of w_4.

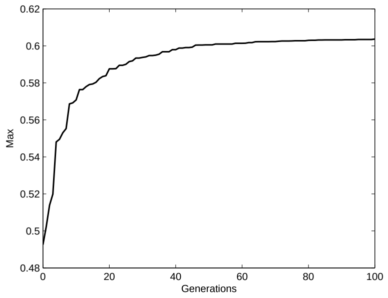

Fig. 5: Convergence profiles for determining upper bounds of w_1.

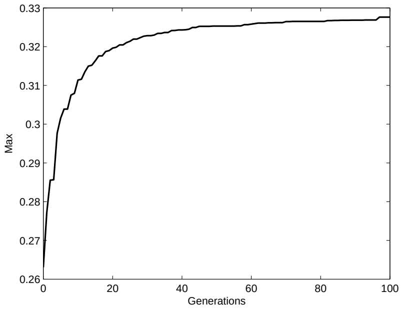

Fig. 6: Convergence profiles for determining upper bounds of w_2.

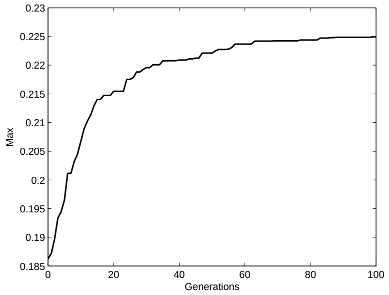

Fig. 7: Convergence profiles for determining upper bounds of w_3.

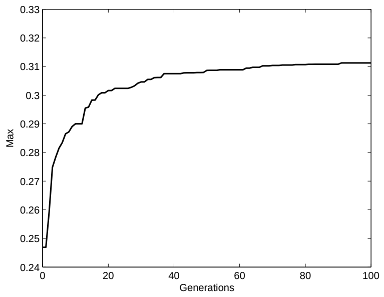

Fig. 8: Convergence profiles for determining upper bounds of w_4.

From Fig.1~8, it can be seen the proposed algorithm has better convergence. In the first 10 generations, the convergence curve has a sharp slope, which means good convergence. From generations 10 to 40, the algorithm converges with high speed and then trends to keep stable. From the statistical results on the data in the immune memory, the interval range of generation 40th has reached 90% of genetation 100th. During generation 60~100, the convergence curve keeps stable and even a rather horizontal level. And the comparison between Fig.1,2,3,4 and Fig.5,6,7,8 shows the convergence of the low bound reaches stable more quickly, probably because the variation range of the low bound is limited.

5. Conclusion

In this paper, we discuss a method for determining weights in interval judgments. We view the computation of the weight vector of the interval number matrix as a nonlinear optimization model with constraints, and propose an immune evolutionary algorithm to optimize the computational process of determining interval weights. The experiment results show that the algorithm has the promising performance of convergence and accuracy.

Acknowledgments

This work was supported in part by the Key Project of the National Nature Science Foundation of China (No. 60534020), Program for New Century Excellent Talents in University from Ministry of Education of China (No. NCET-04-415), the Cultivation Fund of the Key Scientific and Technical Innovation Project from Ministry of Education of China (No. 706024), and International Science Cooperation Foundation of Shanghai (No. 061307041).

References

[1] T. L. Saaty, *The Analytic Hierarchy Process* , McGraw-Hill, New York, 1980.

[2] T.L. Saaty and L.G. Vargas, Uncertainty and Rank Order in the Analytic Hierarchy Process. *European Journal of Operational Research*, 32: 107-117, 1987.

[3] A. Arbel and L.G. Vargas, Preference Simulation and Preference Programming: Robustness Issues in Priority Deviation. *European Journal of Operational Research*, 69:200-209, 1993.

[4] R. Islam, M.P. Biswal, and S.S. Alam, Preference Programming and Inconsistent Interval Judgments. *European Journal of Operational Research*, 97:53-62, 1997.

[5] M. Kress, Approximate Articulation of Preference and Priority Derivation–a Comment. *European Journal of Operational Research*, 52:382-383, 1991.

[6] Y.M. Wang and K.S. Chin, An Eigenvector Method for Generating Normalized Interval and Fuzzy Weights. *Applied Mathematics and Computation*, 181:1257-1275, 2006.

[7] Y.M. Wang and T.M.S. Elhag, A Goal Programming Method for Obtaining Interval Weights from an Interval Comparison Matrix. *European Journal of Operational Research*, 177 : 458-471, 2007.

[8] Y.M. Wang, On Lexicographic Goal Programming Method for Generating Weights from Inconsistent Interval Comparison Matrices. *Applied Mathematics and Computation*, 173:985-991, 2006.

[9] Y.M. Wang, J.B. Yang, and D.L. Xu, A Two-stage Logarithmic Goal Programming Method for Generating Weights from Interval Comparison Matrices. *Fuzzy Sets and Systems* , 152:475-498, 2005.

[10] J.R. Yu and S.J. Cheng, An Integrated Approach for Deriving Priorities in Analytic Network Process. *European Journal of Operational Research* , 180 :1427-1432, 2007.

[11] O.S. Vaidya and S. Kumar, Analytic Hierarchy Process: An Overview of Applications. *European, Journal of Operational Research* , 169:1-29, 2006.

[12] K. Sugihara, H. Ishii, and H. Tanaka, Interval Priorities in AHP by Interval Regression Analysis. *European Journal of Operational Research* , 158:745-754, 2004.

[13] Y.S. Ding and L.H. Ren, Artificial Immune System: Theory and Application. *Pattern Recognition and Artificial Intelligence(in Chinese)*, 13: 52-59, 2000.

[14] Y.S. Ding, A nonlinear PID Controller Based on Fuzzy-tuned Immune Feedback Law. *The 3rd World Congress on Intelligent Control and Intelligent Automation*, pp.1576-1580, 2000.

[15] G.H. Fried, H. George, G.J. Hademenos , and J. George, *Shaum's Outlines of Theory and Problems of Biology(Second Edition)*, McGraw-Hill, New York, 1998.