International Journal of Security and Its Applications
Vol.8, No.3 (2014), pp. 355-362
http://dx.doi.org/10.14257/ijsia.2014.8.3.35

Optimization Study on Interval Number Judgment Matrix Weight Vector Based on Immune Evolution Algorithm

Jianming Sun[1] and Jing Wu[2]

[1]School of computer and information engineering, Harbin University of Commerce, Harbin Postal 150028,China
[2]School of Experiment, Harbin University of Commerce Harbin Postal 150028,China
[1]sjm@hrbcu.edu.cn, [2]wujing@hrbcu.edu.cn

Abstract

This paper focuses on the approach to solve the immune evolution optimization with interval number judgment matrix weight vector. In light of the features of interval number judgment matrix, we transform the solution problem of weights into the optimization one of nonlinear with restriction, design immune evolution algorithm on the base of immune mechanism, and use the convergence ability of immune system to search the optimal access to get interval numbers from matrix weight vector. Moreover, we construct immune operator and form vaccine and optimization strategies by priori knowledge of target problem, by which we can keep individual diversity and elite individuals and abandon useless individuals as soon as possible. In the evolution process, search can be dramatically improved by overcoming the problems of earliness and degradation during global search. Simulation results of experiments show the advantages of this algorithm in accuracy, convergence, convergence rate and so on.

KeyWords: *immune evolution, interval number, matrix weight, nonlinear optimization*

1. Introduction

As is stated above, in the process of risk evaluation of information security, information security is relative to organization management. Different organizations have different opinions on safety attributes such as confidentiality, integrity, serviceability, authenticity, controllability. However, organizations often give undue importance to identifying key assets and key risk factors. Therefore, identifying various index weights scientifically must be solved in evaluating risks of information security.

Until now, there are many approaches to identifying weights [1], such as deciding weight by experts, deciding weight by AHP, deciding weight by entropy values. However, deciding weight by experts depends on experts' knowledge and experience, so this approach is very subjective and random; deciding weight by entropy values is objective and precise by using information in various index and calculating information entropy to fix weight according to importance of various index, but this accuracy is difficult to achieve in the real world. Among these approaches, AHP is widely applied to in many field, because this approach adjusts to qualitative and quantitative analysis and policy making by deconstructing hierarchy of target system and restricting experts' judgment consistently through mean comparison matrix between structural factors. Experts' experiments, knowledge, instinct and so on play a great role in deciding weight by AHP, but risk evaluation of information security is associated with a complicated issue such as organization management, technology and social environment. Thus, experts only give a fuzzy extent-an interval number to show judgment-to judge problems and fix evaluation data because experts cannot be clear about problem's essence and and experiment and

ISSN: 1738-9976 IJSIA

instinct. Obviously, interval number adapts to human judgment more. Whereas, traditional AHP is not good at interval number judgment. That is why it should be improved in order to adjust to decide weight of various indicators in the process of risk evaluation of information security. In this part, we use interval number to construct experts judgment matrix to fit to uncertainty of reality and fuzziness of experts' experiment, explore a new solution approach to weight vector with interval number judgment matrix by means of optimization calculation theory of immune evolution.

2. Classic ahp to Solve Judgment Matrix Weight

This approach is based on classic AHP and combine it into solving judgment matrix weight. Saaty proposed a basic solution approach to judgment matrix weight vector, which includes judgment matrix assignment, eigenvector acquisition and consistently testing one or more circulation.

2.1. Judgment Matrix and its Scale Selection

Under the assumption of a group of factors $A_1, A_2, \cdots A_n$, matrix can be shown according to relatively important degree of pairwise comparison:

$$A = (a_{ij})_{n \times n} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix} \text{,in which } i, j \in [1..n] \quad (1)$$

In this matrix, a_{ij} means importance scale of factor A_i to factor A_j compared to evaluation goal, and pairwise comparison is qualitative. To quantize as far as possible in making policy, Saaty introduces 1-9 scale method to fix value for types of relative important relationship and to transform qualitative evaluation into quantitive evaluation. Table 1 lists values of all scales and their implication.

Saaty selects 1-9 scale method on the basis of following facts and scientific proof:

(1) In the real world, when factors used to compare are the same or close, qualitative difference is significant and precise.

(2) Five types of intensity of importance (equal importance, somewhat more important, much more important, very much more important, and absolutely more important) can fix values appropriately. If higher precision is needed, we can compare adjacent judgments. Therefore, it is appropriate for 1-9 scale method to compare intensity of importance. More distinction of importance can be listed in the scope of 1-9 scale by means of division and clustering. Hence, 1-9 scale is consistent and operative.

(3) 1-9 scale correspond to psychological limit of 7±2.

(4) Social results show people at most need 7 scales to distinguish importance difference as a general rule.

Table 1. Judgment Matrix Scales and its Explanation

Intensity of importance	Explanation
1	Two factors contribute equally to the objective
3	Experience and judgement slightly favour one over the other
5	Experience and judgement strongly favour one over the other
7	Experience and judgment very strongly favour one over the other. Its importance is demonstrated on practice.
9	The evidence favouring one over the other is of the highest possible validity.
2,4,6,8	When compromise is needed
multiplicative inverse	Compare factor A_i with A_j to get a_{ij}, that is, $A_j : A_i = 1/a_{ij}$

Apparently, after fixing values of judgment matrix by 1-9 scale method, we can get positive reciprocal matrix $A = \left(a_{ij}\right)_{n\times n}$, and three conditions in Formulation (2) are needed as follows:

1) $a_{ij} \succ 0$

2) $a_{ii} = 1$;

3) $a_{ji} = 1/a_{ij}$ (2)

2.2. Acquisition of Weight Vector

If λ represents maximum characteristic root of A, and w represents characteristic vector of A corresponding to λ, vector of w can be decided by Formulation (3)

$$Aw = \lambda w \tag{3}$$

If $i, j, k \in N$ and $a_{ij} = a_{ik}a_{kj}$, A can be called consistency judgment matrix. According to features of consistent matrix, consistent matrix must be positive reciprocal matrix. On the contrary, it is not. In constructing judgment matrix, we must get positive reciprocal matrix by 1-9 scale method, as it is shown in Formulation (2). However, consistent conditions are not necessarily satisfied because of the influence of experts' knowledge and preference. Hence, it is necessary to testify consistency in order to ensure accuracy and reliability of judgment.

Saaty uses Formulation (4) to examine consistency of judgment matrix.

$$CR(A) = \frac{\lambda - n}{(n-1)RI} \le 0.1 \tag{4}$$

If consistent conditions in Formulation (4) can be satisfied, A can be considered as satisfaction consistency matrix. Otherwise, A will not characterize satisfaction consistency. In Formulation (4), RI means average random consistent indicator, its value can be shown in Table 2:

Table 2. Value of RI (Average Random Consistent Indicator)

Order n of Judgment Matrix	1	2	3	4	5	6	7	8	9	10
RI	0	0	0.58	0.90	1.12	1.26	1.36	1.41	1.46	1.49

If Formulation (4) can be satisfied, judgment matrix can accept inconsistency; otherwise, judgment matrix which is built at the beginning is not satisfactory and needs reassigning and rectifying until consistency in this matrix can be found. The solutionof weight vector in judgment matrix is based on the process in which Formulation (4) adjust judgment matrix appropriately to satisfy consistency examine. This part makes foundation on it and builds optimization model and algorithm of solving interval number judgment matrix weight vector on the basis of immune evolution mechanism.

3. Optimization Model and Immune Evolution Algorithm of Solving Interval Number Weight Vector

3.1. Demonstration of Interval Number of Judgment Matrix and its Weight Vector

Interval number can be demonstrated by $X^I = \left[\underline{x}, \overline{x}\right]$, in which $\underline{x}$ and $\overline{x}$ respectively represent interval number's lower limit and upper limit. If $\underline{x} = \overline{x}$, X^I will reduce to a real number. Interval number is actually a way to reveal fuzzy knowledge. Interval number conforms to ordinary binary operation rule. Assume $a^I = [\underline{a}, \overline{a}]$ and $b^I = [\underline{b}, \overline{b}]$, the rules of binary operation that interval number should conform to can be demonstrated by [3]:

(1) $a^I + b^I = [\underline{a} + \underline{b}, \overline{a} + \overline{b}]$;

(2) $a^I \times b^I = [\underline{a} \times \underline{b}, \overline{a} \times \overline{b}]$;

(3) $a^I \div b^I = \left[\frac{\underline{a}}{\overline{b}}, \frac{\overline{a}}{\underline{b}}\right]$,especially, $\frac{1}{[\underline{a}, \overline{a}]} = \left[\frac{1}{\overline{a}}, \frac{1}{\underline{a}}\right]$;

(4) $ka^I = [k\underline{a}, k\overline{a}], \forall k \succ 0$;

Combine it to Saaty's 1-9 scale method, and interval number judgment matrix $A^I = (a_{ij}^I)_{n \times n}$ ($i, j \in N$) can be shown by:

(1) $a_{ii}^I = 1, i = 1, 2 \cdots, n$;

(2) $a_{ij}^I = [\underline{a_{ij}}, \overline{a_{ij}}], \frac{1}{9} \le \underline{a_{ij}} \le \overline{a_{ij}} \le 9, \forall i, j$;

(3) $a_{ij}^I = \frac{1}{a_{ij}^I}, \forall i, j$.

Obviously, corresponding weight vectors are also interval number, which can be shown by $w^I = (\underline{w_i}, \overline{w_i})_{n \times 1}$.

3.2 Optimization Model of Immune Evolution

Sugihara [4] summarizes the features of interval number matrix as followings: If $A^I = [\underline{A}, \overline{A}]$ ($\underline{A} = (a_{ij})_{n \times n}$, $\overline{A} = (\overline{a_{ij}})_{n \times n}$) and $\underline{\lambda}$ and $\overline{\lambda}$ respectively represent maximum characteristic roots of $\underline{A}$ and $\overline{A}$, $\lambda = [\underline{\lambda}, \overline{\lambda}]$ is characteristics interval number of A^I.

On that basis, we inherit the above approach of solving judgment matrix. In the light of satisfaction consistency conditions in Formulation (4), this part will introduce a solution model of interval number weight vector, which can be demonstrated as followings: ,

We assume $M = \{1, 2, \cdots, m\}$. A^I which means interval number judgment matrix can generate m judgment matrix $A^i, i \in M$, then we solve maximum principal characteristics and corresponding weight vector of each A^I , finally get weight vector $w^i, i \in M$.

To $w_j^j, j \in N$, we can get lower limit and upper limit, which can be respectively demonstrated by $\underline{w_j} = \min\{w_j^i | i \in M\}$ and $\overline{w_j} = \max\{w_j^i | i \in M\}$, and then we can know interval number weight vector of A^I is $w_I = (w_j^I)_{n \times 1} = ([\underline{w_j}, \overline{w_j}])_{n \times 1}$. The value of component area of w^I is decided by weight vector of judgment matrix $A^i, i \in M$ which is from any combination of values of A^l's component area. Therefore, the problem of solving interval number weight vector can be transformed into the one of optimization with restriction like demonstration of Formulation (5) and (6).

$$\text{P1} \quad \underline{w_j} = \min\left\{ w_j^i \middle| i \in M \right\}$$
$$\text{s.t.}$$
$$A^i w^j = \lambda^i w^j,$$
$$CR\left(A^i\right) \le 0.1,$$
$$i \in M. \tag{5}$$

$$\text{P2} \quad \underline{w_j} = \max\left\{ w_j^i \middle| i \in M \right\},$$
$$\text{s.t.}$$
$$A^i w^j = \lambda^i w^j,$$
$$CR\left(A^i\right) \le 0.1,$$
$$i \in M. \tag{6}$$

Apparently, solution in Formulation (5) and (6) is a problem of complicated optimization. To acquire weight vector as fast as possible, it is key to construct $A^i, i \in M$, that is to say we need to traverse space of certainty judgment matrix which conform to interval number, and realize fast convergence to lower and upper limit of judgment matrix weight vector which meet restriction. In this paper, we use the global convergence ability of immune system to solve interval number of weight vector by means of immune evolution algorithm. In immune system, antibody can testify antigen and assist immune cells to eliminate antigen. Once antibody tests antigen, antibody can generate clonal expansion and then produce lots of antibody. After that, new antibody will experience high frequent mutation. With the elimination of antigen, antibody library of dynamic expansion will remove some antibody with lower affinity to antigen and then achieve stability of antibody group. At the same time, predominant antibody will be remain by immune memory and be able to induce higher secondary immune response. As far as optimization issue mentioned in this part is concerned, target function is antigen while judgment matrix decided randomly is antibody. Through generating all kinds of immune operators and keeping antibody and immune memory, we can find antigen and then solve weight vector of interval number judgment matrix as fast as possible.

4. Design of Algorithm

This part will introduce the specific steps of immune evolution algorithm that is used to solve interval number judgment matrix weight. The steps are demonstrated by Table 3.

Some concepts and operators related to this algorithm can be explained as followings:

(1) Antibody encoding: matrix encoder is used to antibody and encoder results match with judgment matrix in form, that is to say they have same dimensions. Components of matrix will be encoded by real number, but numeric area of each component is decided by interval number. This encoded mode can fulfill fully traverse matrix space decided by interval number judgment matrix.

(2) Affinity function: In this part, we take day beacon function as affinity function, while satisfaction consistency conditions are taken as elimination operators to optimally eliminate antibody.

(3) Clonal expansion: In antibody group, firstly select dominant antibody according to adaptive value. Selection ratio is m_{select}. Sub-antibody can be produced by single-pont arithmetic crossover between every two-antibody. Clonal coefficient is m_{clonal}.

Arithmetic crossover refers to two real numbers a and b can generate $c: c = m_c a + (1 - m_c) b$ by crossover, in which $0 \le m_a \le 1$ is produced randomly and is called arithmetic crossover parameter.

(4) Hypermutation: Single-point hypermutation can be implemented to antibody from antibody group which is achieved by clonal expansion. If mutation probability is m_{hyperm} , and lower and upper limit of elements of real number encoding are respectively L and U , $c = L + m_{hm}(U - L)$ can be acquired by mutation.

(5) Negative selection: To antibody group by clonal expansion, we select and eliminate some antibody with lower affinity by which antibody group can keep stable.

(6) Immunological memory is essentially a elite strategy, which can preserve dominant antibody.

(7) When algorithm meets terminating conditions, that is evolution algebra gets to maximum m_{gen} , we can stop searching.

Table 3. Algorithm Steps

Steps	Explanation
Step 1:	Initialize interval number judgment matrix and parameters of all algorithm;
Step 2:	Initialize antibody population (scale of population is m_{pop});
Step 3:	Select dominant antibody to implement clonal expansion;
Step 4:	Implement hypermutation to antibody group which has been expanded;
Step 5:	Eliminate antibody with lower affinity by negative selection to keep it stable;
Step 6:	Keep dominant antibody by immunological memory mechanism;
Step 7:	When algorithm meets terminating conditions, searching will be over and then output dominant antibody as antigen, otherwise, searching will turn to Step 3 and repeat from Step 3 to Step 7.

Take interval number judgment matrix from Reference [158] as an example. Numerical value can be demonstrated by:

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

We utilize the above immune evolution algorithm and parameter setting in Table 4, then we can acquire calculation results shown in Table 5. To analyze the results easily, we list the results of reference [7] in Table 4.

Table 4. Parameter Setting

Parameter	Value	Parameter	Value
Scale of antibody population (m_{pop})	40	iteration algebra of algorithm (m_{gen})	100
Coefficient of clonal selection	0.25	Coefficient of clonal expansion	2
Probability of hypermutation	0.5		

Table 5. Results of Solving Interval Number Judgment Matrix Weight

Results of algorithm in this paper			Results from reference[5]		
weight	lower limit	upper limit	weight	lower limit	upper limit
w_1	0.342	0.605	w_1	0.369	0.552
w_2	0.141	0.327	w_2	0.150	0.290
w_3	0.090	0.224	w_3	0.093	0.189

According to Sugihara, interval number judgment matrix weight vector should meet the requirement of Formulation (7) and (8),

$$\sum_i \overline{w}_i - \max_j \left(\overline{w}_j - \underline{w}_j \right) \geq 1 \tag{7}$$

$$\sum_i \underline{w}_i + \max_j \left(\overline{w}_j - \underline{w}_j \right) \leq 1 \tag{8}$$

It is shown in Formulation (7) and (8) that we should try to expand searching range of components of each weight when to solve interval number judgment matrix weight vector. In this way, it is more suitable to express uncertainty of interval number weight. As Table 5 is shown, the results of this part meet Formulation (7) and (8), which means this algorithm is correct and valid. Comparing them with the results of reference [7], we can find the results of this part better reflect scope that uncertainty of weight distribution can get to.

In this experiment, convergence curve of each component of weight vector can be demonstrated by from Fig1 as followings.

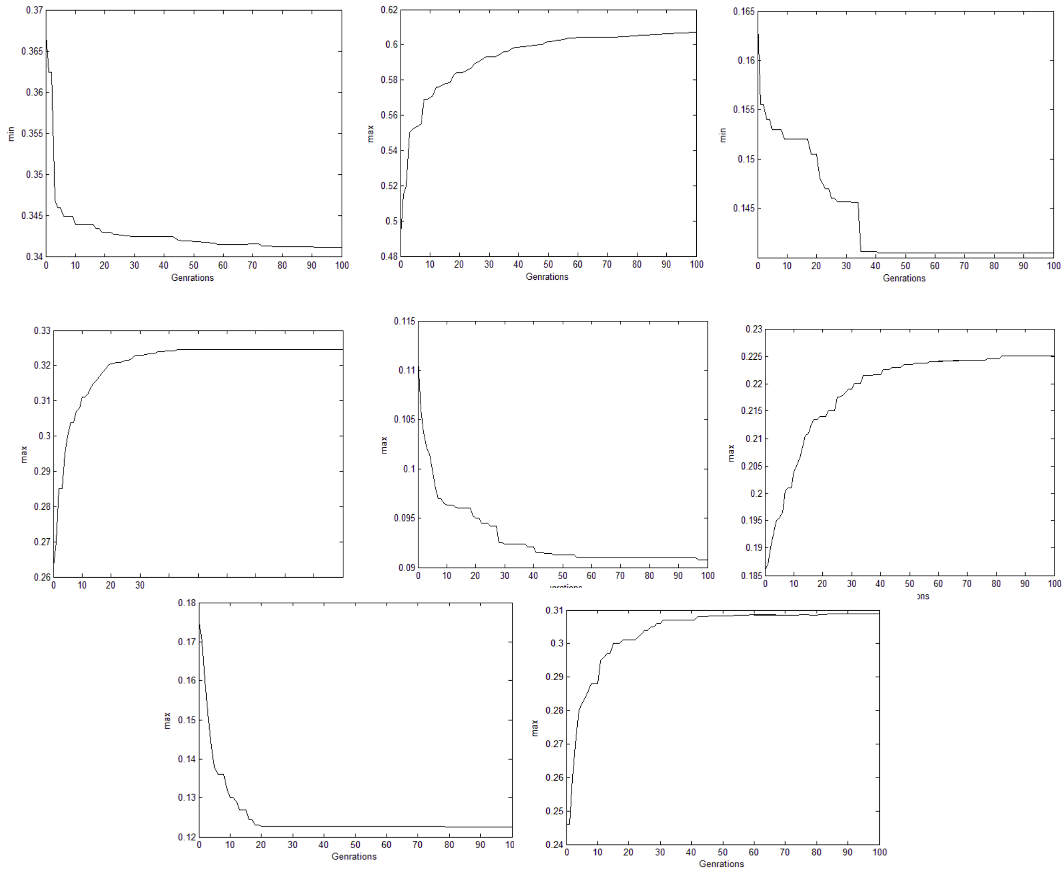

Figure 1. Convergence Curve of each Component of Weight Vector (a) Convergence Curve of Solving Lower Limit of *w1* ,(b) Convergence Curve of Solving Upper Limit of *w1*,(c) Convergence Curve of Solving Lower Limit of *w2* ,(d) Convergence Curve of Solving Upper Limit of *w2*(e) Convergence Curve of Solving Lower Limit of *w3* ,(f)Convergence Curve of Solving Upper Limit of *w4* ,(g)Convergence Curve of Solving Lower Limit of *w4* ,(h)Convergence Curve of Solving Upper Limit of *w4*

5. Conclusion

We can find that algorithm mentioned in this part has very good convergence. In the first 10 generations after initialization, slope of convergence curve is very big, which reflects algorithm in the first 10 generation has high convergence. However, from Generation 10 to Generation 40, curve has faster convergence rate and gradually keeps stable. Based on statistics in immunological memory, interval number in Generation 40 is 90% of Generation 100. From Generation 60 to Generation 100, curve keeps very steady; what's more, convergence curve of most testing cases keeps horizontal. Comparing Chart 1,3,5,7 with Chart 2;4;6;8, we know that convergence curves of lower limit is more inclined to keep stable than those of upper limits. That is because interval number is bigger than zero, and change scope of lower limit is restricted.

Acknowledgements

This work was supported by The Education Department of Heilongjiang province science and technology research projects (Grant Nos. 12531160).

References

[1] X. Zeshui and S. Zaidong, "Priority Method for a Kind of Multiattribute Decision-making Problems", Journal of Management Sciences in China, vol. 5, no. 3, **(2002)**.
[2] K. Sugihara, H. Ishii and H. Tanaka, "Interval Priorities in AHP by Interval Regression Analysis", European Journal of Operational Research, vol. 158, no. 3, **(2004)**.
[3] D. Yongsheng, "The New Framework of Computational Intelligence: Biological Network Structure", Caai Transactions on Intelligent Systems, vol. 2, no. 2, **(2007)**.
[4] D. Yongsheng and R. Lihong, "Design of a Bio-network Architecture Based on Immune Emergent Computation", Control and Decision, vol. 18, no. 2, **(2003)**.
[5] A. Arbel and L. G. Vargas, "Preference Simulation and Preference Programming: Robustness Issues in Priority Derivation", European Journal of Operational Research, vol. 269, no. 2, **(1993)**.
[6] L. Jiaomin, L. Yongzheng, M. Junying and W. Zhenzhou, "Intrusion Detection System Alerts Fusion Based on Fuzzy Comprehensive Evaluation and Immune Evolution", Journal of Computational Information Systems, vol. 9, no. 4, **(2013)**.
[7] A. R. Yildiz, "A new hybrid differential evolution algorithm for the selection of optimal machining parameters in milling operations", Applied Soft Computing, vol. 13, no. 3, **(2013)**.
[8] G. Maoguo, J. Licheng and Z. T. Lining, "Baldwinian learning in clonal selection algorithm for optimization", Information Sciences, vol. 180, no. 8, **(2010)**.