

## A co-evolutionary algorithm integrated with immune multi-agent

Jianhong Ma<sup>1,a</sup>, Han Zhang<sup>1,b\*</sup>, Baofeng He<sup>2,c</sup>

<sup>1</sup> Software Technology School, ZhengZhou University, 450002, China

<sup>2</sup> Department of Computer Science, Sias International University, Zhengzhou University,  
451150, China

<sup>a</sup>majianhong@zzu.edu.cn, <sup>b</sup>tianyadehui@tom.com, <sup>c</sup>hnddhibf@sina.com

**Keywords:** immune; co-evolution; evolutionary algorithm; multi-agent; affinity

**Abstract.** In this paper we present a co-evolutionary algorithm integrated with immune multi-agent based on analyzing the lack of the immune evolutionary algorithms and co-evolutionary algorithm. First, antibody deputies are generated through the competition among populations and within populations sort and based on the calculation target of antibody information entropy and affinity, then the algorithm searches the optimal solution is implemented through affinity maturation and Gene library update evolution. The simulation results show that the algorithm has a good search capability of the optimal solution, calculation accuracy and convergence rate is superior to the traditional algorithm.

### 1 Introduction

Immune evolutionary algorithm<sup>[1-3]</sup> is a new optimization algorithm of simulated evolution in the biological immune system and genetic processes. Its essence lies in the record of the best individual information was formed after the clone variation, and the optimal individual evolution was instead of population evolution. Immune evolutionary algorithm based on random search method of probability theory, which can effectively prevent the algorithm premature fall into local optimal value. But because of the characteristics of the search direction is random and population evolution has the characteristics, the overall performance of the immune evolutionary algorithm remains to be studied<sup>[4]</sup>. Coevolutionary genetic algorithm (CGA) as an optimization algorithm of rapid development, is an extension of the traditional evolutionary algorithm. The algorithm of the model consists of two or more populations. Different population co-evolution in the ecosystem, and interaction, finally makes the ecological system evolution<sup>[5]</sup>. CGA has been widely used in many fields<sup>[6-7]</sup>. The cooperative co-evolutionary algorithm using individual and population collaboration, competition or cooperation way and evaluation of individual information based on interactions between individuals, so that the algorithm has a better search space size of the global search for solutions of better<sup>[8]</sup>. It is difficult for the existing models of cooperative co-evolution to effectively produce collective complex adaptive cooperation behaviors because of their deficiency in solving problems with interaction between sub systems. This algorithm has the imperfect place, aiming at the problem of solution in the overall dimension segmentation, the process of algorithm can not be better reflected in the collaborative thinking, how to best individual species in another species affect the optimal solution search, still need to be studied<sup>[9]</sup>.

The co-evolution model fusion immune mechanism proposed by article<sup>[10]</sup>. When the fitness of the population decreases the model took the elite antibody guidance strategy to guide. But the elite antibody on each agent were optimal matching antigen affinity or not, need

real-time updating calculation. When the difference between multiple agent had this guide would lead to lower actuarial accuracy. When the difference between multiple agent this guide would lead to lower actuarial accuracy. A new immune cooperative multi-agent model is proposed in literature, The mechanism of antibody represents extraction is used in sorting the affinity, and for multiple Agent make antibodies did not campaign research, and the algorithm is not given a crossover operator and mutation operator in detail, when the algorithm is run to after a certain period of time, synergistic degradation is required when a reasonable strategy to guide how to carry on the processing algorithm. Based on the above research, this paper proposes an immune co-evolutionary algorithm based on multi-agent, experiments show that the algorithm has good convergence speed and calculation accuracy.

## 2 Co-evolutionary algorithm design

### 2.1 Definition

**Definition1** Antibody  $Agent_i \in Agent (S, V_i) (1 \leq i \leq n)$ , Among them  $Agent$  is the set of antibody,  $Agent_i$  corresponds to the solution space in the  $S$  subset of  $V_i$ , namely the solution environment.

**Definition2** Antibodies represent  $Rep_i (1 \leq i \leq n)$  is the set of all antibodies represent in the current iteration. For any  $Agent_i$ , Antibodies represent  $Rep_i$  of  $Agent_j (1 \leq i \leq n, j \neq i)$  corresponding the antibody to be identified.  $Agent_i$  self antibody corresponding anti-idiotypic antibody.

**Definition3** If  $Agent_i$  has  $m$  antibodies, each antibody gene number is  $n$ . Its code is  $s = \{s_1, s_2, \dots, s_{|s|}\}$ ,  $|s|$  is the set size of code symbol. Information entropy  $H(j)$  of genetic unit  $j$  is:

$$H(i) = - \sum_{j=1}^{|s|} p_{ij} \log_2(p_{ij}) \quad (1)$$

In formula (1),  $p_{ij}$  is the probability the  $i$ -th coded symbols on the  $j$ -th gene unit in set  $s$ . That is  $p_{ij} = \frac{n_{ij}}{m}$ ,  $n_{ij}$  is the total number of the  $i$ -th coded symbols in the  $j$ -th gene unit.

### 2.2 Algorithm design

#### 2.2.1 The basic idea of the algorithm

Each iteration calculation, inter  $Agent_i$  uses antibodies represent extraction and antigen presenting two processes interact. Antibody representatives from antibody in  $Agent_i$  set, information of  $Agent_j$  are treated by  $Agent_i$  antibodies as anti-idiotypic antibody. The antibody treatment idiotypic antibody is immune recognition process. The process of Antigen presentation is idiotypic antibody sets mapping to display antigen set before immune recognition. Immune recognition and antibody extraction is the core process consisting of affinity calculation algorithm, affinity maturation and gene library evolution. The process realized immune recognition and extraction of antibodies represent each iteration. In the process of identification, collection antibody and gene library iterative evolution, each update optimal solving environment, output for solving the problem of optimal solution finally.

#### 2.2.2 The key steps of the algorithm described

##### (1) Initialization

This step mainly includes the initial assignment antibody set and gene library. Initialize the gene pool is generally determined by the domain model of knowledge base. In general, genetic unit consists of constraints conditions or components of solving the problem, Then alleles in gene unit are mapped to the gene pool. The initial we use gene recombination, i.e. selection of allelic

composition antibodies from the gene pool. Assuming that t moment, antibody in s position, The probability of allele j is chosen to be  $p(s, j, t)$ , that is

$$p(s, j, t) = \frac{c(s, j, t)}{\sum_{i=1}^k c(s, j, t)} \prod_{i=1}^w a_i * h_i(s, j, t) \quad (2)$$

In formula (2), gene concentration is  $c(s, j, t)$ , The heuristic function  $h_i(s, j, t)$  of the weight coefficient is  $a_i$ .

(2) Competition in the generation of antibodies represent

Competition in the generation of antibodies represent include  $Agent_i$  of internal competition and competition between populations. For  $Agent_i$  of internal competition, according to a descending sort affinity, is divided into two parts of  $Agent_k(o)=[a_1', a_2', \dots, a_o']$  and  $Agent_k(o)=[a_{o+1}', a_{o+2}', \dots, a_{o+p}']$ . Among them  $o=p=m/2$ ,  $a_1'$  is the greatest affinity,  $a_{o+p}'$  is the smallest affinity. When we conduct antibody extraction, we select antibody in  $Agent_k(o)$  inter population competition.

For inter population competition, we first find the information entropy in each  $Agent_i$  antibody. For arbitrary  $Agent_i$ , the average information entropy of its antibody is:

$$H = \frac{1}{n} \sum_{j=1}^n H(j) \quad (3)$$

So the k-th antibody in  $Agent_i$  for competitiveness  $J(k)$  is:

$$J(k) = l * \left[ \frac{maff(k)}{\frac{1}{nn} \sum_{i=1}^m maff(i)} + \frac{H(k)}{\frac{1}{nn} \sum_{i=1}^m H(i)} \right] \quad (4)$$

In formula (4),  $maff(k)$  is all antibody affinity mean in  $Agent_i$ ,  $H(k)$  as the average information entropy,  $l$  as its weight coefficient. The adaptive assignment strategy is:

$$l = l * \left( 1 + \frac{T}{T_m} * c \right) \quad (5)$$

In formula (4),  $c=0.01$ ,  $T$  is the number of iterations,  $T_m$  is the largest number of repeated. When  $T=1$ ,  $l=1$ .

Competitive reward coefficient is  $Cb_k$ , of  $Agent_i$ , thus

$$Cb_k = L * J(K) \quad (6)$$

In formula (6),  $L$  is the amplification coefficient, when  $J(k)$  increases, the greater the  $Cb_k$ . When evolutionary process tends to equilibrium, the competitiveness of  $Agent_i$  value had little difference, there will be harmonious ability decline phenomenon. Therefore, evolutionary algorithm runs to a certain extent, will need to each  $Agent_i$  according to competitive values in descending order, and change the  $Cb_k$ :

$$Cb_k = L * J(K) + H_k \quad (7)$$

In formula (7),  $J(k)$  of Sub populations  $Agent_i$  is more big, ranking the in front, the greater the competitive reward value  $H_k$ .

(3) Antigen presentation

$Agent_i$  selected  $R$  elements from  $Rep[j]$  to add antigen set Antigens.

(4) Antibody affinity calculation

For the domain model  $i$  existence  $\text{Match}(a,b)$ , it can calculate the matching values between antibody  $a$  and antigen  $B$ . Each  $\text{Agent}_i$  uses the same matching function.  $\text{Affinity}(a)$  depends on the degree of antigen in  $\text{Agent}_i$  and Antigens.  $\text{Affinity}(a)$  can be mapped as objective function in the optimization model. With the  $S$  values of the average maximum value in  $R$   $\text{Match}(a,b)$  is  $\text{Affinity}(a)$ . Mature antibody set  $\text{Matured}_i(t)$  is:

$$\text{Matured}_i(t) = \left\{ ab \mid \text{Affinity}(ab) \geq \frac{1}{\text{size}} \sum_{j=1}^{\text{size}} \text{Affinity}(ab_j) \wedge ab \in \text{Antibody}_i(t) \right\} \quad (8)$$

##### (5) Affinity maturation

Affinity maturation consists of cloning, mutation and compression of three parts. Suppose that the  $J$ -th of the mature antibody in  $\text{Matured}_i(t)$  is  $ab_{ij}$ , then the clone size  $q_{ij}$  is:

$$q_{ij} = \text{Int} \left[ \text{size} * \text{countsize} * \frac{\text{Affinity}(ab_{ij})}{\sum_{s=1}^{|\text{Matured}_i(t)|} \text{Affinity}(ab_{is})} \right] \quad (9)$$

In formula (9),  $\text{countsize}$  is a clone of the size of the total.

When the variation, we use mutation operator to search the neighborhood of  $ab_{ij}$   $q_{ij}$  times. Selection of mutation operator depends on the character of the problem, evaluation and selection updates of single mutation efficiency is get on according to the solution of the problem. We choose anti idiotypic variation to conduct mutation operation in this article.

Assuming the sub population of the current antibody is  $Ab_i = \{e, ab_1, ab_2, \dots, ab_m\}$ ,  $ab_j$  is the antibody which need mutation operation. The mutation individuals randomly chosen is  $\langle ab_{k_1}, ab_{k_2} \rangle$ , and  $j, k_1, k_2 = 1, 2, \dots, m$ . When the mutation probability of  $ab_j$   $p_m < p$ , that is:

$$Ab_j' = ab_j + ab_j * m * (|ab_{k_1}| - |ab_{k_2}|) \quad (10)$$

In formula (10),  $|ab_{k_1}| - |ab_{k_2}|$  is antibody difference,  $ab_j * m$  is antibody coefficient of  $ab_j$ , that is:

$$ab_j * m = \frac{e * f - ab_j * f}{e * f - ab_{\min} * f} \quad (11)$$

In formula (11),  $e$  is competition antibody of Subgroup species of  $Ab_i$ ,  $ab_{\min}$  is the minimum fitness value antibody in  $Ab_i$ ,  $f$  is the antibody fitness value.

Compression strategies is to mature antibody  $ab_{ij}$ , the antibody after variation

$$d_{ij} = \max \{ \text{Affinity}(ab_s) \mid s = 1, 2, \dots, q_{ij} \} \quad (12)$$

In formula (12), when  $\text{Affinity}(ab_{ij}) < \text{Affinity}(d_{ij})$ , the antibody original  $ab_{ij}$  is updated by  $d_{ij}$ .

After affinity maturation process,  $\text{Matured}_i(t)$  into  $\text{Matured}_i'(t)$ ,  $\text{Antibody}_i(t)$  into  $\text{Antibody}_i'(t)$ , their conversion of set is the same of scale.

##### (6) Gene library evolution

Hypothetical gene concentration rate value is natural attenuation of  $\rho$  parameters which makes the probability in the current iteration expression gene concentration better was chosen at the next iterative evolution is larger, then:

$$c(x, y, t+1) \leftarrow \rho * c(x, y, t) + (1 - \rho) * \Delta c(x, y, t) \quad (13)$$

$$\Delta c(x, y, t) = \begin{cases} \text{Affinity}(ab_x) & y \in ab_x \mid \wedge ab_x \in \text{Matured}_i'(t) \\ 0 & \text{otherwise} \end{cases} \quad (14)$$

In formula (13,14),  $1 \leq x \leq L, 1 \leq y \leq K$ .

##### (7) The iterative termination

Conditions on the composition algorithm terminates is constituted by algorithm a maximum number of iterations or the maximum running time condition.

### 3 Experimental simulation and analysis

#### 3.1 The test function

In order to verify the performance of solving function optimization of the co-evolutionary algorithm integrated with immune multi-agent, compared with the genetic algorithm and clonal selection algorithm. During the experiment, we choose 8 different types of functions for algorithm testing, these 8 functions are divided into two groups of single peak and multi-peak. Table 1 and table 2 is the function expression.

Table 1 Unimodal functions and expressions.

The test function	The function expression	intervals	optimal value
$f_1$	$f_1(x) = \sum_{i=1}^n x_i^2$	$[-100, 100]^{30}$	0
$f_2$	$f_2(x) = \sum_{i=1}^n  x_i  + \prod_{i=1}^n  x_i $	$[-100, 100]^{30}$	0
$f_3$	$f_3 = \sum_{i=1}^n \left( \sum_{j=1}^i x_j \right)^2$	$[-100, 100]^{30}$	0
$f_4$	$f_4 = \sum_{i=1}^{n-1} \left[ 100(x_{i+1} - x_i)^2 + (x_i - 1)^2 \right]$	$[-30, 30]^{30}$	0

Table 2 Multi-peak functions and expressions

The test function	The function expression	intervals	optimal value
$f_5$	$f_5(x) = \sum_{i=1}^n (-x_i * \sin \sqrt{ x_i })$	$[-500, 500]^3$	-12569.5
$f_6$	$f_6(x) = \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i) + 10)$	$[-5.12, 5.12]^3$	0
$f_7$	$f_7(x) = \sum_{i=1}^n \frac{x_i^2}{4000} - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$[-600, 600]^3$	0
$f_8$	$f_8(x) = \frac{1}{n} \sum_{i=1}^n (x_i^4 - 16x_i^2 + 5x_i)$	$[-100, 100]^{30}$	-78.33236

#### 3.2 Simulation and analysis

In order to analyze the influence of the algorithm parameters, the number of iterations was chosen as index. The comparison algorithm we choose is genetic algorithm optimization configuration OGA and clonal selection algorithm CSA in this paper. The optimization results are compared. Table 2.  $f_5$  and  $f_8$  were chosen to test the peak function. In the iterative process, antibody mutation probability and cross promultibability according to the fitness value and the concentration of adaptive are adjusted.

Design parameters of antibody population number GN in the interval [2, 10] was sampled 1 as the step length. The termination of the experiment condition setting is  $|f_{\text{best}} - \text{OPT}| < \epsilon$ , NP=300, The number of iterative algorithms is 2000, then the algorithm fitness changes as shown in Figure 1.

In figure 1 star line is GN=4, the square line is GN=8. Figure(a) shows the  $f_5$  function changes of the fitness, figure(b) shows the  $f_8$  function of the fitness changes. we can know that the antibody population number GN has a larger effect on the performance of the algorithm from the figure. When GN value is small in the function the change trend of fitness value is slow. When the GN value increases, the convergence speed and precision of algorithm gradually get better.

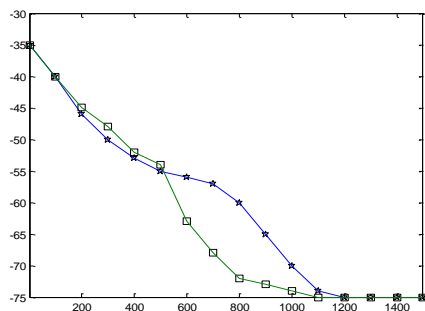


Fig 1. Different GN values  
algorithm fitness changes(a)

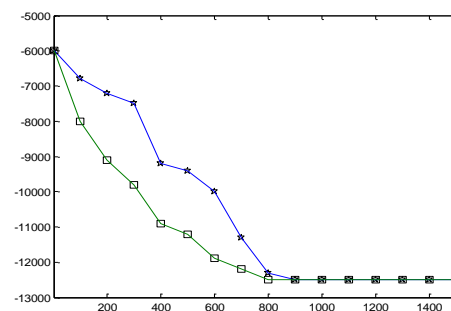


Fig 1. Different GN values  
algorithm fitness changes(b)

## 4 Conclusions

Through the defects of the analysis of the immune evolutionary algorithm and the cooperative co-evolutionary algorithm, The present situation of solving some problems cannot be reasonably decomposed into multi-swarm to decomposition optimization and merge, we propose an immune co-evolutionary algorithm based on multi-agent. The algorithm mainly depends on the competition among and within populations to generate antibody representative. It takes the sub population information entropy and individual antibody affinity as competitive computing strategy. Antibody competition rewards index adjustments when evolved to a certain stage, the algorithm avoids degradation collaborative computing capabilities. Anti idotype mutation operator to subgroup variation can improve the algorithm's global search ability. Experimental analysis shows that, the algorithm has good convergence speed and calculation accuracy.

## Acknowledgments

We thank all the anonymous reviewers for their valuable comments. The project is supported by Science and technology projects (142102210500) of Henan province; Henan province (122102210518); Henan provincial key research projects (15A520029)

## References

- [1] Liu Xing-bao, Cai Zi-xing. Hybrid immune evolutionary algorithm for global optimization problems[J]. Journal of xidian university, 2010, 37(5): 971-980.
- [2] Wang Lei, Jiao Licheng, Wang Yong, Peng Wei-xiong. Immune Evolutionary Algorithms[C]//Proc. of IEEE 5th International Conference on Signal Processing. [S.l.]: IEEE Press, 2002.
- [3] Ni Chang-jian, Ding Jing, Li Zuo-yong. Immune Evolutionary Algorithm[J]. Journal of Southwest Jiaotong university, 2003, 38(2): 87-91.
- [4] Guo Zhong-quan, Luo Wen-cai, Chen Xiao-qian. Research on multi-method collaborative immune evolution-ary algorithm[J]. Computer Engineering and Applications, 2012, 48(14): 17-22.
- [5] Zhang Yun-kai, Wang Fang-Wei, Zhang yu-qing. Co-evolutionary Genetic Algorithm and Its Applications[J]. Computer Engineering, 2004, 30(15): 38-43.
- [6] H. Chen, K.P. Wong, D.H.M. Nguyen, and C.Y. Chung, "Analyzing oligopoly electricity market using coevolutionary computation," IEEE Trans. Power Syst., vol. 21, no. 1, pp. 143–152, Feb. 2006.

- [7]F.Careri,C.Genesi,P.Marannino,M.Montagna,S.Rossi,and I.Siviero,"Strategic bidding in a day-ahead market by coevolutionary genetic algorithms,"in Proc.IEEE Power Energy Soc.Gen.Meeting, Jul.25-29,2010,pp.1-8.
- [8]Dong Hongbin,Huang Houkuan,Yin Guisheng,and He Jun.An Overview of the Research on Coevolutionary Algorithms[J].Journal of Computer Research and Development,2008,45(3): 454-463.
- [9]LIU Xiaoyong, LIU Jun, ZHAO Guirong, et al. Study on cooperation evolution model[J].Computer Engineering and Applications,2012, 48(8):51-54.
- [10]Mu CH,Jiao LC,Liu Y.M-Elite coevolutionary algorithm for numerical optimization[J].Journal of Software,2009,20(11):2925 2938.