An Improved Immune Algorithm and Its Evaluation of Optimization Efficiency

Chengzhi Zhu, Bo Zhao, Bin Ye, and Yijia Cao

College of Electrical Engineering,
National Laboratory of Industrial Control Technology,
Zhejiang University, Hangzhou 310027, Zhejiang, China
{yijiacao, zhuchengzhi}@zju.edu.cn

Abstract. Based on clonal selection principle, an improved immune algorithm (IIA) is proposed in this paper. This algorithm generates the next population under the guidance of the previous superior antibodies (Ab's) in a small and a large neighborhood respectively, in order to realize the parallel global and local search capabilities. The computational results show that higher quality solutions are obtained in a shorter time, and the degree of diversity in population are maintained by the proposed method. Meanwhile, "Average truncated generations" and "Distribution entropy of truncated generations" are used to evaluate the optimization efficiency of IIA. The comparison with clonal selection algorithm (CSA) demonstrates the superiority of the proposed algorithm IIA.

1 Introduction

The natural immune system is a complex but self organizing and highly distributed system. It employs a multilevel defense against invaders through nonspecific (innate) and specific (acquired) immune mechanisms. The natural immune system is a subject of great research interest because of its powerful information processing capabilities. In particular, it performs many complex computations in a completely parallel and distributed fashion [1]. Over the last few years, there are many application areas in which immunity-based models appear to be very useful [2]. An immune genetic algorithm has been proposed by Cao et al. to solve packing problem effectively [3]. Gao has applied the immune algorithm to the power network planning, and compared it with that based on genetic algorithm. The results show that the immune algorithm is better than the genetic algorithm in global optimization [4]. Furthermore, Timmis et al. have employed the artificial immune system in the knowledge discovery of database, and compared it with the normal K Means and Kohonen network [5]. The artificial immune system has been expected strong advantage over those conventional methods to the field of information processing.

The defensive mechanisms of the natural immune system are very effective, and can be used as a source of inspiration for computation problems. If the objective function of the practical problem and its solution are respectively regarded as the invading antigen (Ag) and the antibody (Ab) generated by the immune system, the solving process of the practical problem is quite similar to the natural immune mechanisms. In 2001, based on

L. Wang, K. Chen, and Y.S. Ong (Eds.): ICNC 2005, LNCS 3611, pp. 895 – 904, 2005.

the clonal selection principle, De Castro and Von Zuben proposed a general clonal selection algorithm, named CLONALG (in this paper, we call it CSA) [6]. The clonal selection principle is used to explain the basis features of an adaptive immune response to an antigenic stimulus, which establishes the idea that only those cells recognizing the Ag's are selected to proliferate. The algorithm was derived primarily to carry out machine-learning, pattern-recognition task and optimization problems. Nevertheless, when solving complex function optimization tasks and some engineering optimization problems, the algorithm has the low degree of diversity in population, and may converge to a local optimum. So a reliable global approach would be of considerable value to intelligence and computation community.

In this paper, an improved immune algorithm (IIA) is proposed through modifying the cell clone and hypermutation mechanism. The cell clone operation proliferates the clonal selected B cells in a small neighborhood to produce the Ab's with high affinity, thereby improving the local search capabilities of the algorithm and obtaining better solution. While the hypermutation operation mutates the selected cells in a large neighborhood to improve the global search capabilities and maintain the diversity of the population. To demonstrate the superiority of the method, simulation results for four benchmark functions have been compared with various techniques available in literature, namely, standard genetic algorithm (SGA), clonal selection algorithm (CSA) and particle swarm optimization algorithm (PSO). Meanwhile, the optimization efficiency of IIA is evaluated. It is shown that the optimization efficiency of IIA is higher than that of CSA.

2 Clonal Selection Theory and Improved Immune Algorithm

2.1 Clonal Selection Theory

Learning in the immune system involves raising the population size and affinity of those lymphocytes that have proven themselves to be valuable by having recognized any Ag's. Any molecule that can be recognized by the adaptive immune system is known as an Ag. When an animal is exposed to an Ag, some subpopulation of its B cells responds by producing Ab's. By binding to these Ab's and with a second signal from accessory cells, such as the T cell, the Ag stimulates the B cell to proliferate (divide) and mature into terminal Ab secreting cells, called plasma cells. The process of cell division (mitosis) generates a clone. B cells, in addition to proliferating and differentiating into plasma cells, can differentiate into long-lived B memory cells. Memory cells commence to differentiate into plasma cells capable of producing high-affinity Ab's, when exposed to a second antigenic stimulus. The main features of the clonal selection theory are [6]:

i. Proliferation and differentiation on stimulation of cells with Ag's. The selected Ab's are cloned independently and proportionally to their antigenic affinities, generating a repertoire of clones. The higher the antigenic affinity, the higher the number of clones generated for each of selected Ab's.
ii. Generation of new random genetic changes, expressed subsequently as diverse Ab patterns, by a form of accelerated somatic mutation. (a process called affinity maturation); the higher the affinity, the smaller the mutation rate.

iii. Estimation of newly differentiated lymphocytes carrying low-affinity antigenic receptors.

2.2 Improved Immune Algorithm

In the implementation of the proposed algorithm, the Ag's and Ab's represent the optimization problems and their candidate solutions respectively. While the fitness functions of the candidate solutions are regarded as the antigenic affinities of the Ab's. The algorithm comprises the following five operators, clonal selection, cell clone, hypermutation, receptor editing and elitist preserving.

Firstly, the clonal selection operation selects the B cells with the highest-affinity Ab receptors to be the B memory cells. Secondly, the cell clone process proliferates the clonal selected B cells in a small neighborhood to produce the Ab's with high affinity, which can improve the local search capabilities of the algorithm and obtain better optimal solution. Thirdly, the hypermutation process is performed dependent on receptor affinity. Cells with low-affinity receptors may be further mutated and, as a rule, die if they do not improve their clone size or antigenic affinity. In cells with high-affinity Ab receptors, however, hypermutation may become inactive, generally in a gradual manner. Unlike the CSA, the IIA undergoes hypermutation in a relatively large neighborhood to realize the global exploration, which may rescue the solving stuck on unsatisfactory local optima. Then, to improve the diversity of the population and escape from local optima ulteriorly, receptor editing is performed. Those B cells undergone receptor editing delete their low-affinity receptors and developed entirely new ones in the feasible region randomly. Finally, the elitist preserving strategy is adapted to maintain the convergence of the algorithm in each iteration processing. [7]

The procedure of the IIA can be described as follows. For the convenience of description, no distinction is made between a B cell and its receptor, known as an Ab.

Step 1: Parameters definition: the radius of the cell clone r, the radius of the hypermutation R, the number of the populations, the maximum generation Gen_{max}, the initial population A_t, that composed by M random initial Ab's, t=0.

Step 2: t=t+1.

Step 3: Clonal selection: Select N ($N \le M$) highest affinity Ab's from A_t to compose a new set B_t of memory Ab's. This paper defines N=int (α*M), where α is selection probability, and $0<\alpha<1$.

Step 4: Cell clone: IIA generates randomly (M-N) new sets of memory Ab's B_t' in a relative small neighborhood around the Ab's in B_t. Suppose $x_{i,t}$ is an Ab in B_t, Then, the new Ab generated randomly is:

$$x_{i,t}^{'} = \begin{cases} x_{i,\min} & x_{i,t} * rand(1-r,1+r) < x_{i,\min} \\ x_{i,\max} & x_{i,t} * rand(1-r,1+r) > x_{i,\max} \\ x_{i,t} * rand(1-r,1+r) & otherwise \end{cases}, \quad (1)$$

where r is the radius of cell clone, and $r \in [0,1]$; $[x_{i,min}, x_{i,max}]$ is the space of feasible region; $rand(1\text{-}r, 1+r)$is a random number between $1\text{-}r$ and $1+r$.

As shown above, the number of clones is proportional to the antigenic affinity. Then, a selecting method is proposed based on the "roulette wheel" strategy in this paper. Suppose the antigenic affinities of the N Ab's in B_t is $f(1),f(2),\ldots,f(N)$ respectively, then the probability of a new Ab generated for each of the Ab's in B_t is:

$$p(k) = \frac{f(k)}{\sum_{i=1}^{N} f(i)}; \qquad k = 1,2,\ldots,N \,. \tag{2}$$

Define $S(0) = 0$, then

$$S(k) = \sum_{i=0}^{k} p(k); \qquad k = 1,2,\ldots,N \,. \tag{3}$$

Generate $(M\text{-}N)$ random numbers evenly distributed between 0 and 1, $\xi s \in U(0, 1)$, $s = 1,2,\ldots,M\text{-}N$. If $S(k\text{-}1) < \xi s < S(k)$, then select Ab k as a new Ab generated randomly in the neighborhood around the "mother Ab", whose radius is r. According to this method, generate $M\text{-}N$ new Ab's. So, the higher the antigenic affinity, the higher the number of clones generated for each of the selected Ab's, which means the algorithm has more chance to explore new Ab with higher affinity in a small neighborhood of the elitist. In a word, the cell clone is a process to search for the local optimum.

Step 5: Hypermutation: Cell clone is merely a process to search for the local optimum. However, to prevent from unexpected local optima, and meanwhile, to obtain the ability to search for the global optimum, IIA undergoes the process of hypermutation, in which random genetic changes are introduced into each of the Ab's in a large neighborhood. Suppose $x_{i,t}$ is an Ab in B_t, Then, the new Ab after mutation is:

$$x_{i,t}^{''} = \begin{cases} x_{i,\min} & x_{i,t} * rand(1-R,1+R) < x_{i,\min} \\ x_{i,\max} & x_{i,t} * rand(1-R,1+R) > x_{i,\max} \\ x_{i,t} * rand(1-R,1+R) & otherwise \end{cases}, \tag{4}$$

where R is the radius of cell clone, generally. R is larger than the radius of cell clone r; $rand(1\text{-}R,\ 1+R)$ is a random number between $1\text{-}R$ and $1+R$. The process of hypermutation generates a set $B_t^{''}$ composed of N new Ab's.

Step 6: Compose a new C_t of the new Abs generated by the cell clone and hypermutation processes. Then $C_t = B_t^{'} \cup B_t^{''}$, and M is the number of the new Abs.

Step 7: Receptor editing: To improve the diversity of the population and escape from local optima ulteriorly, receptor editing is performed in IIA. Those B cells replace the d lowest affinity Ab's from C_t by the d new Ab's in set D_t. Where $d=int\ (\mu*M)$; μ is the editing probability, and $0<\mu<1$.

Step 8: Elitist preserving: The B cells replace the lowest affinity Ab's from D_t by the Ab's with the highest affinity in set A_t, and form a new set A_{t+1} of Ab's. The algorithm maintains its convergence based on elitist preserving.

Step 9: If the termination condition is satisfied, then the iteration processing stops, else, go to step 2.

2.3 Parameters Setting

Studies of IIA for function optimization have indicted that good performance requires proper values of the parameters. A low probability of selection may result in convergence to local optima. On the contrary, a high probability of selection may result in poor speed of convergence. Typical values of selection probability are in the range 0.4~0.5. The value of the editing probability can not be too large either. Although a high probability of selection may improve the diversity of the algorithm, it can decrease the efficiency of convergence. Based on the experience, typical values for selection probability are in the range 0.1~0.2.

The radius of hypermutation R and the values of the radius of clone selection r determine the ability to converge in the global solving space and explore the optimum in the local neighborhood. They are both set according to the optimization problems. Generally, R is ten to fifty times of r. The value of r decreases linearly as follows:

$$r = r_{\max} - \frac{r_{\max} - r_{\min}}{Gen_{\max}} \times Gen \,, \tag{5}$$

where $Gen_{\max}$ is the maximum generation set in IIA, and Gen is the current generation.

On the early stage of the searching process, the value of r is set to be relatively large to maintain the global search capability. Then, on the latter stage, the solutions gradually move close to the optimum, therefore, small value of r is needed to realize local searching. This method to determine the value of r does improve the performance of the IIA.

3 Evaluation Criterion of Optimization Efficiency

In order to evaluate IIA's convergence speed and degree of instability, this paper presents two indices, "Average truncated generation" and "Distribution entropy of truncated generation". Thereafter, they are unified as a monolithic criterion.

Definition 1: Truncated generation
A global numerical optimization can be formulated as solving the following objective function:

$$\max f\left(x_i\right), \quad i=1,2,\ldots,n, \quad s.t.\, a_i \le x_i \le b_i \,. \tag{6}$$

Based on one of the possible strategies (such as different mutation probability and selection probability), when the computing accuracy ε ($\varepsilon = f_{max} - f$) is reached. The final generation is defined as the truncated generation. If the computing accuracy would not be reached until the predefined maximum generation Gen_{max}, then the truncated generation is defined as Gen_{max}.

Definition 2: Average truncated generation
Assume that the algorithm is performed for L runs, and T_i is defined as the truncated generation of the i *th* run, then the set T is composed of the T_i as follows: $T = \left\{ T_i \,\middle|\, 0 < T_i \le Gen_{\max},\ T_i \in Z^+,\ i = 1,2,\ldots,L \right\}$.

Arranging the elements in the set T according to their magnitudes, then a new set $T' = \left\{ T_i' \middle| T_i' < T_{i+1}', \ i = 1,2,..,K-1, K \le L \right\}$ can be derived. Given $C = \left\{ C_i \middle| 0 < C_i \le L, \ C_i \in Z^+, \ i = 1,2,..,K \right\}$ and $P = \left\{ p_i, \middle| p_i = \frac{C_i}{L}, \ \sum p_i = 1, \ i = 1,2,..,K \right\}$, when the algorithm reaches its computing accuracy ε ($\varepsilon = f_{max} - f$) under the guidance of strategy S, the average truncated generation is defined as follows:

$$T(S,\varepsilon) = \sum_{i=1}^{K} T_i' p_i, \tag{7}$$

Definition 3: Distribution entropy of truncated generation
The distribution entropy of truncated generation can be defined as follows:

$$H(S,\varepsilon) = \frac{-\sum_{i=1}^{K} T_i' \ln(p_i)}{\ln(K)}, \tag{8}$$

when the algorithm reaches its computing accuracy ε ($\varepsilon = f_{max} - f$) based on strategy S. The definition of K and P_i is same as that defined in definition 2. The distribution entropy of truncated generation represents the measure of uniformity that the distribution of the truncated generation have and the stability of the algorithm.

According to the above definitions, the average truncated generation is used to evaluate the average convergence speed of the optimization algorithm for several independent runs. The distribution entropy of truncated generation is used to evaluate whether the convergence of the algorithm is stable. The lower its degree, the more stable the convergence of the algorithm. This paper unified the two indices as a monolithic criterion on the plane (T, H) to evaluate the optimization efficiency of the proposed algorithm based on different strategies [8]. Then on the plane (T, H), the closer point to the origin represents the higher the optimization efficiency.

4 Experiment Results

In this paper, four benchmark functions are given as follows, which are widely used to test the efficiency of the optimization algorithms.

$$F_1 = 0.5 + \frac{\sin^2\sqrt{x_1^2 + x_2^2} - 0.5}{\left[1.0 + 0.001\left(x_1^2 + x_2^2\right)\right]^2}, \qquad |x_i| \le 100$$

$$F_2 = 100\left(x_1^2 - x_2\right)^2 + \left(1 - x_1\right)^2, \qquad |x_i| \le 2.048$$

$$F_3 = \left(x_1^2 + x_2^2\right)^{0.25}\left[\sin^2\left(50\left(x_1^2 + x_2^2\right)^{0.1}\right) + 1.0\right], \qquad |x_i| \le 10$$

$$F_4 = \frac{1}{4000}\sum_{i=1}^{30} x_i^2 - \prod_{i=1}^{30}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1, \qquad |x_i| \le 600.$$

To demonstrate the superiority of the proposed IIA approach, simulation results for the above benchmark functions have been compared with various techniques available in literature, namely, standard genetic algorithm (SGA), clonal selection algorithm (CSA) and particle swarm optimization algorithm (PSO).

The maximal generation is set to Gen_{max} =100 for all four algorithms and the population size is set to be 100. The computing accuracy is set to be 10^{-5}.To avoid any hazardous interpretation of optimization results, related to the choice of particular initial population, we performed the simulation 200 times for each function, starting from different populations randomly generated in the search space. The rest running parameters of the algorithms are chosen to be those by which the best performance could be obtained.

Figure 1 shows convergence characteristic of F_1 obtained using the four optimization algorithms respectively. The 'fitness' shown in the Figure are the average values of the optimal individual in each generation during the 200 runs of each algorithm. It is clear for the figure that the solution obtained by IIA converges to higher quality solutions at earlier iterations (about 15 iterations) rather than the other three algorithms. Similar results can be obtained for F_2 to F_4.

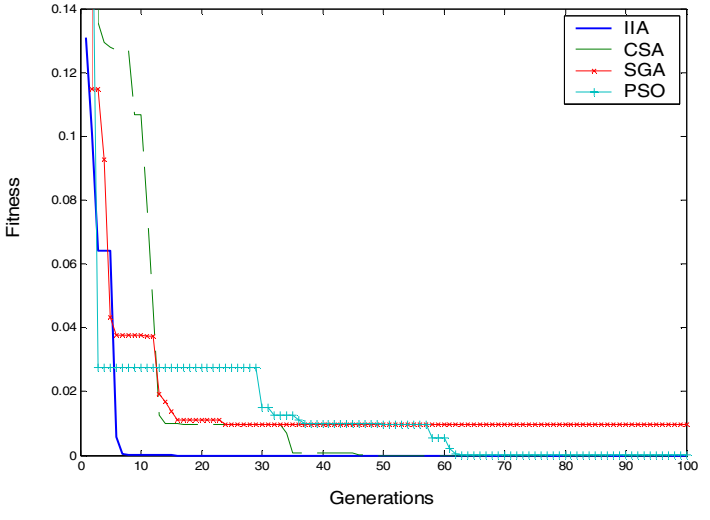

Fig. 1. Optimization procedure with four algorithms

Table 1. Comparison of optimal results for different methods

func	IIA		CSA		SGA		PSO	
	Best	Average	Best	Average	Best	Average	Best	Average
F_1	0	0	0	2.62e-10	0	1.11e-2	9.97e-10	3.90e-3
F_2	5.57e-9	9.21e-5	1.50e-6	8.05e-4	2.57e-7	3.25e-3	2.61e-8	6.45e-4
F_3	0	6.23e-21	0	2.90e-3	1.37e-4	1.26e-2	3.78e-6	5.15e-4
F_4	0	0	0.58	1.26	0.79	2.65	0.57	0.94

Table 2. Comparison of iteration for different methods

func	IIA		CSA		SGA		PSO	
	Best	Average	Best	Average	Best	Average	Best	Average
F_1	11	15	23	58	41	92	59	96
F_2	13	48	18	94	79	97	32	57
F_3	50	61	27	82	100	100	100	100
F_4	45	52	100	100	100	100	100	100

Table I and II summarizes the optimal results and convergence iterations of the best and average solutions as obtained by different methods when applying to the all four benchmark functions over 200 runs. These results show that the optimal solutions determined by the IIA lead to lower optimal value than that found by other methods, which confirms that the IIA is well capable of determining the global or near-global optimum solution. It can also be seen that IIA performs better than other methods in convergence speed.

The phenomenon sufficiently incarnates the characteristics of IIA as follows:

i. Clonal selection and elitist preserving operations both preserve the high-affinity Ab's. This feature makes IIA maintain its convergence.
ii. IIA select high-affinity Ab's to undergo cell clone operation in small neighborhoods, by which the fine search around a local minimum is performed.
iii. Hypermutation are operated in a large neighborhood. Therefore, IIA can improve its global search capabilities.

5 Evaluation of IIA's Optimization Efficiency

This paper performs the F_1 optimization task to evaluate the optimization efficiency of IIA based on a monolithic criterion defined in section 4, which is combined by the two proposed indices "Average truncated generations" and "Distribution entropy of truncated generations". The results are compared with that of CSA.

Meanwhile, the effects of parameters r and R on the optimization efficiency are also evaluated in the following descriptions.

The population size of each algorithm is set to be 100. The IIA and CSA are both processed for 100 generations (Gen_{max} =100) and repeated for 200 runs. The computing accuracy is 10^{-5}. During IIA operation, the radius of cell clone is set to be r_{max}=0.1, r_{min}=0.05, and the radius of hypermutation is R=20, $r_{\min}$=1.

5.1 Selection Probability α

To evaluate the optimization efficiency in relation to α, we fix μ=0.2. And α is assumed as the following values {0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9} respectively. On the plane (T, H) shown in Fig.2, each point represents the result obtained by the optimization algorithm when taking corresponding parameter pair (α, μ).

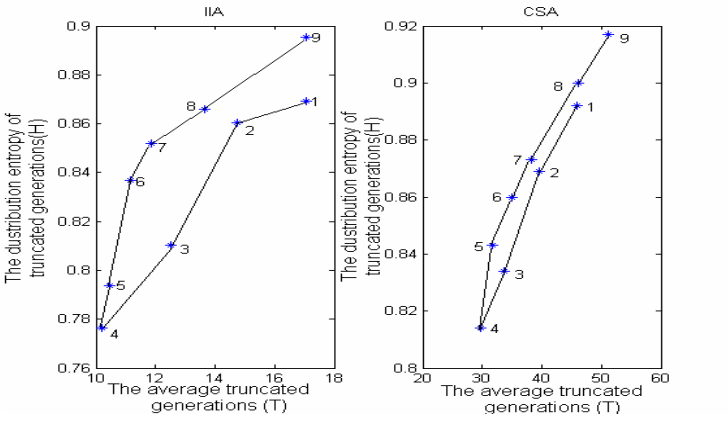

Fig. 2. Optimization efficiencies with various selection operators

From Fig.2, we can see that with the increase of α, the optimization efficiency rises gradually to the maximum when α=0.4, and then decreases.

5.2 Editing Probability μ

In order to study how μ effects on the optimization efficiency, α is fixed to be 0.4, while μ takes various values {0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45} respectively.

Fig.3 shows the similar results to those in Fig.2: The optimization efficiency rises gradually to the maximum when μ=0.2, and then decreases, with the increase of μ.

Moreover, both in Fig.2 and Fig.3, each point in the left plane (*T, H*) is closer to the origin than the corresponding one in the right plane, which demonstrate that with the proper parameters, the proposed algorithm have higher optimization efficiency than CSA.

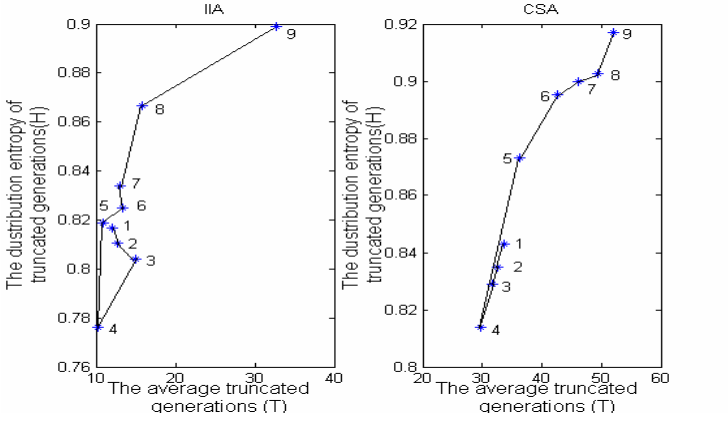

Fig. 3. Optimization efficiencies with various editing operators

6 Conclusions

This paper proposes an improved immune algorithm based on clone selection principle. IIA contributes mainly to introducing two new operators, cell clone and elitist preserving, meanwhile, modifying the hypermutation operator. Therefore, the parallel global and local searching capabilities can be obtained. Simulation results for some benchmark functions show that IIA greatly outperforms the algorithms SGA, PSO and CSA.

Acknowledgements

This work is supported by the Outstanding Young Scholars Fund (No. 60225006) and Innovative Research Group Fund of Natural Science Foundation of China.

References

1. Li, T.: Computer Immunology. Beijing: Publishing House of Electronics Industry (2004)
2. Jiao, L. C., Du, H. F.: Development and Prospect of the Artificial Immune System. Acta Electronica Sinica. 31 (2003) 1540-1548
3. Cao, X. B., Liu, K. S., Wang, X. F.: Solving Packing Problem Using an Immune Genetic Algorithm. Mini-Micro System. 21 (2000) 361-363
4. Gao, J.: The Application of the Immune Algorithm for Power Network Planning. System Engineering – Theory & Practice. 21 (2001) 119-123
5. Timmis, J., Neal, M., Hunt, J.: Data Analysis Using Artificial Immune Systems, Cluster Analysis and Kohonen Networks: Some Comparisons. Proc. IEEE SMC '99 Conference (1999) 922-927.
6. De Castro, L. N., Von Zuben, F. J.: Learning and Optimization Using Clonal Selection Principle. IEEE Trans. Evol. Comput. 6 (2002) 239-251
7. Zuo, X. Q., Li, S. Y.: Adaptive Immune Evolutionary Algorithm. Control and Decision. 19 (2004) 252-256.
8. Sun, R. X., Qiu, L. S.: Quantitative Evaluation of Optimization Efficiency for Genetic Algorithms. Acta Automatica Sinica. 26 (2000) 552-553