■ 647

A Self-adaptive Multipeak Artificial Immune Genetic Algorithm

Qingzhao Li[1] , Fei Jiang[2,3*]
[1] School of Foreign Languages, Suzhou University, Suzhou 234000, Anhui, China
[2] Laboratory of Intelligent Information Processing, Suzhou University, Suzhou 234000, Anhui, China
[3] School of Information Engineering, Suzhou University, Suzhou 234000, Anhui, China
*Corresponding author, e-mail: feiwuhan@126.com

Abstract

Genetic algorithm is a global probability search algorithm developed by simulating the biological natural selection and genetic evolution mechanism and it has excellent global search ability, however, in practical applications, premature convergence occurs easily in the genetic algorithm. This paper proposes an self-adaptive multi-peak immune genetic algorithm (SMIGA) and this algorithm integrates immunity thought in the biology immune system into the evolutionary process of genetic algorithm, uses self-adaptive dynamic vaccination and provides a downtime criterion, the selection strategy of immune vaccine and the construction method of immune operators so as to promote the population develop towards the optimization trend and suppress the degeneracy phenomenon in the optimization by using the feature information in a selective and purposive manner. The simulation experiment shows that the method of this paper can better solve the optimization problem of multi-peak functions, realize global optimum search, overcome the prematurity problem of the antibody population and improve the effectiveness and robustness of optimization.

Keywords: Genetic Algorithm, Artificial Immune System, Multipeak Function

1. Introduction

With the rapid development of information technology, intelligent computation methods have been applied in an increasing number of fields and the widely-used intelligence algorithms include genetic algorithm, ant colony algorithm and immune algorithm and so on. Genetic algorithm is an evolutionary algorithm with pervasive influence formed at the very first beginning and its basic principle is to exchange information through the genetic operations such as crossover, duplication and mutation in order to obtain the near optimum solution to the optimization problem [1]. Biology immune system is a highly-parallel information learning system. It can self-adaptively recognize and eliminate the antigenical foreign objects from the body and it has adjustment ability such as memory and learning. Inspired by the intelligent behaviors of the biology immune system, artificial immune algorithm is a random search optimization algorithm integrating certainty and randomicity [2]. At present, intelligence algorithms have been widely used in machine learning, pattern recognition, robot behavior simulation, intelligent trouble diagnosis, management science and social science and they are applicable to solve complicated non-linear and multi-dimensional space optimization problems.

In as early as the 1940s, the scholars have been researching how to use the computer to simulate biology technology. In the early 1970s, Professor Holland from Michigan University, USA, has drawn inspiration from biological genetics and evolutionary mechanism and proposed genetic algorithm, which is a search mechanism based on the theory of natural evolution and which is an optimization method applicable for the computation of complicated system; however, genetic algorithm has a poor local search ability. Therefore, how to preserve the excellent individuals and the population diversity has been a difficulty in the genetic algorithm [3]. After the 1970s, people have gradually realized the enlightenment of biology immune mechanism in developing new computational intelligence. As a new algorithm inspired from biology immune system, artificial immune algorithm (AIA) focuses on simulating the immunology with computational and mathematical model. It simulates the biology immune system and constructs the corresponding information processing algorithms by learning its structure,

functions and features. By integrating the ideas of GA and AIA, immune genetic algorithm helps the antibody to continuously search and evolve in the solution space through immune operations and evaluates the matching degree between the antibody and the antigen as well as the similarity between the antibodies based on the affinity until the optimal solution is generated. However, the basic immune genetic algorithm has slow and stagnant search process brought by the static designated crossover, mutation probability and vaccine [4].

This paper firstly describes the principles of basic genetic algorithm and artificial immune algorithm. Then based on the summary of the characteristics and shortcomings of these two algorithms, it improves the algorithm by combining immunology and genetics and proposes an self-adaptive multi-peak immune genetic algorithm. Finally, the numerical simulation of the test function demonstrates that this algorithm can obtain global optimal solution effectively and verifies the effectiveness and practicality of this improved algorithm in the paper so that it provides a solution to the multi-peak function optimization problems.

2. Genetic Algorithm

Genetic algorithm (GA) simulates the reproduction, crossover and genetic mutation in natural selection and natural inheritance and it is a randomized search algorithm by referring to the natural selection and genetic mechanism in the biological world. It takes the encoding of the decision variable as the operand and the objective function value as the search information and adopts multiple search points to perform possibility search. The group of original solution randomly generating from solving the problems with GA is called population and the problem-solving process starts from the sub-set. Each individual of the population is a solution to the problem and the encoding string obtained from performing binary coding to the individual is called chromosome. The chromosome is measured by the fitness. The fitness function is a function to measure the environmental adaptation of every individual and its value is the main basis to realize "survival of the fittest". According to the fitness, select certain individuals from the parent generation and the offspring generation and generate the next generation of chromosome population which is more self-adaptive to the environment. After certain generations, the algorithm converges to the optimal solution or the second-optimal solution to the problem [5]. To map the optimal solution space to GA space is indicated as Figure 1.
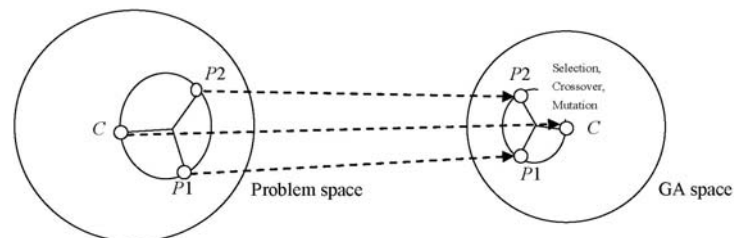

Figure 1. Map the optimization problem space to GA space

Selection, crossover and mutation are the main three operators in GA and they simulate the survival of the fittest and the genetic process.

(1) Selection

The first step of selection is to calculate the fitness. Every individual in the set to be selected has a selection probability, which depends on the fitness and distribution of the individual in the population and uses the probability ratio of the fitness of every individual to determine the possibility to preserve its offspring. Assume that the population size is N and the fitness of the individual i is f_i, then the probability P_s for the individual i to be selected is:

$$P_s = f_i \Big/ \sum_{i=1}^{N} f_i \qquad (i = 1, 2, \cdots, N) \tag{1}$$

Obviously, the individual with a higher selection probability has a higher possibility to be selected and the probability to be inherited to the population in the next generation is directly proportional to the fitness of this individual. Proportional selection is also called Roulette Wheel Selection. It divides the roulette into N portions based on the selection probability P_s. When rotating the roulette in the selection, if a certain point falls into the ith sector, then select the individual i. The distribution ratio of the fitness value of the Roulette Wheel Selection is indicated as Figure 2.
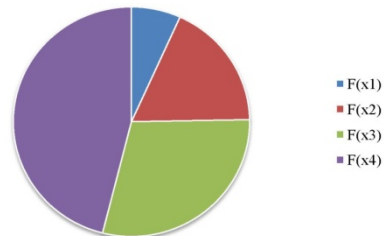

Figure 2. Distribution ratio of fitness value of roulette wheel selection

(2) Crossover

Crossover is to exchange the genes in the same position in two different individuals to be used to reproduce the next generation and generate new individuals [6]. In other words, it takes the two individuals from the population after selection operation as the crossover objects, randomly generates a crossover point and perform crossover on some genes in the right side of the crossover point, as indicated in Figure 3.
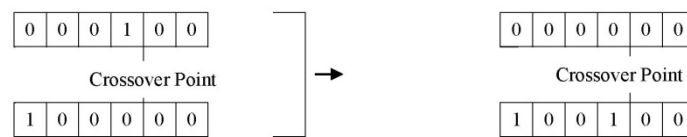

Figure 3. Crossover operator

(3) Mutation

Mutation is to replace the gene value of certain locus in the encoding string of the individual chromosome with other allele gene in the same locus, produce mutation at a small probability or step length and form a new individual. The simplest way is to change the numerical value of a certain position in the string. For the binary encoding, interchange 0 and 1: change 0 to 1 and 1 to 0.

It is an auxiliary method to generate new individuals and it determines the local search ability of GA. The combination of mutation operator, selection operator and crossover operator helps GA have the local random search ability and prevent local convergence. Generally speaking, low-frequency mutation can avoid the possible missing of the important and single gene. The commonly-used mutation operation includes basic bit mutation, uniform mutation and binary mutation [7].

Genetic algorithm is easy for parallelization processing and its fitness function is not restricted by such conditions as continuity and differentiation and it has an extensive application scope; however, GA is easy to get trapped in local optimal solution and its global search ability is also very limited.

3. Artificial Immune Algorithm

Biological immune system has the features such as immune memory, antigen recognition and preservation of antibody diversity. By learning from the above-mentioned features, immune algorithm generates a computation model by integrating engineering

application. Comparing the immune algorithm with the common search methods to solve optimization problems, the affinity between antigen and antibody as well as between antibody and antigen correspond to the matching between the objective function of the optimization problem and the optimal solution as well as between the solution and the objective function. Artificial immune algorithm includes such operations as vaccine extraction, vaccine inoculation and vaccine selection [8, 9]. The steps to solve optimization problems with immune algorithm are classified as follows.

Step 1: Identify the problem, take the given objective function and constraints as the antigen of the algorithm, namely to extract the effective information of the problem to be optimized, the objective function of the optimal problem and the description of the constraints.

Step 2: Generate the antibody population, randomly generate the original antibody population of a certain scale. The antibody population is represented with binary encoding and it defines the antibody as the feasible solution of the objective function in the constraints.

Step 3: Generate immune memory cell, calculate the direct affinity between each antibody and antigen respectively as well as the similarity between the antibodies and preserve the antibody with a bigger fitness value as the memory cell.

Step 4: Select the antibody and calculate the affinity, calculate the concentration of the antibodies with similar fitness in the current antibody population as well as the affinity or fitness of every solution. Decrease the selection probability of the individual with high concentration and increase the selection probability of the individual with low concentration.

Step 5: Differentiate the memory cells, sort them based on the affinity, put the antibody with high affinity to the antigen into the memory cell, take the first few antibodies to form the memory warehouse and preserve the current optimal feasible solution.

Step 6: Evolve and update the antibody with crossover and mutation operators, including the antibody acceleration and suppression and generate new antibody population. Replace the individual of low fitness value with that of high fitness value in the memory cell and form the next-generation antibody population. Eliminate the antibodies with low affinity of the antigen and high concentration and form a new generation of population.

Step 7: Judge whether the termination conditions are satisfied, namely to find the optimal solution or to reach the maximum iterations. If so, output the optimal solution; otherwise, continue Step 3.

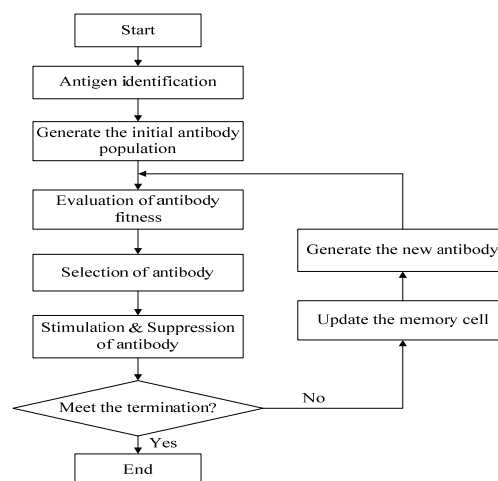The basic flowchart of artificial immune algorithm is indicated as Figure 4.

Figure 4. Flowchart of basic artificial immune algorithm

4. Process of Self-adaptive Multipeak Immune Genetic Algorithm

Artificial immune system can more better ensure that algorithm is not easy to fall into local optimal solution due to its unique antibody diversity maintaining mechanism during the search process. The main idea is to adopt the dynamic extraction vaccine method to avoid the

disadvantage of slow convergence speed resulted from the effectiveness of static vaccine in the process of evolution. The selection update mechanism based on the concentration is adopted to replace the selection replication based on the fitness in terms of the original genetic algorithm, thus maintaining the diversity of solution group. The self-adaptive adjusting crossover probability and mutation probability guides the search process towards the global optimization [10, 11]. The steps of algorithm in this paper are as follows:

Step 1: Encode in view of characteristics of questions to be solved with the purpose to generate the original population. Set such parameters as the crossover probability p_c, the mutation probability p_m, the population size N, the binary encoding length L, the max evolution generation T, the medium individual number of genetic operation M, and the concentration threshold value ε.

Step 2: Take the given objective function and constraint condition as the antigen of questions to be solved to generate original antigen group. Randomly generate N individual populations and extract M individuals from memory warehouse to form the original antigen group, in which, M is the number of individuals in the memory warehouse.

Step 3: Calculate the concentration of antibody c_v.

$$c_v = \frac{1}{N} \sum_{w=1}^{N} ac_{vw} \tag{2}$$

In which: $ac_{vw} = \begin{cases} 1 & ay_{vw} \geq \varepsilon \\ 0 & otherwise \end{cases}$, ε is the determined threshold value. ay_{vw} is the affinity between antibody ab_v and antibody ab_w.

Step 4: Extract vaccines and calculate each individual's fitness value in the current population to find out the optimal individual, and take its all genes as vaccine V. Set M antibodies with fitness optimization in stored population of each generation, then $K*M$ antibodies in total are retained in recent generation K to form the optimal antibody group. The probability of each gene level of each antibody is:

$$p_{i,j} = \frac{1}{K*M} \sum_{j=1}^{K*M} a_j \tag{3}$$

Take k_j with max probability $p_i, i = 1, 2, \cdots, L$ in such allelic gene as the vaccine segment in such allelic gene, thus finally extracting the vaccine $V = (v_1, v_2, \cdots, v_L)$.

Step 5: Vaccination, select antibody to be vaccinated from the previous generation group and generate new immune individuals by exchanging the genetic code values. The selection manner of the genetic segment is as follows:

$$q_i = \frac{p_i}{\sum_{j=1}^{L} p_j} \tag{4}$$

q_i is the probability of each vaccine segment selected and L is the length of the genetic segment. $i, j = 1, 2, \cdots, L$.

Step 6: Population update, the following way is adopted to implement the group updating based on the concentration:

$$p_s(v) = c_v \times \alpha \times \left| 1 - \frac{fit(v)}{Maxfit(v)} \right| + \beta \times \frac{fit(v)}{Maxfit(v)} \tag{5}$$

In which, α and β belongs to adjustable parameters between 0 and 1, and $p_s(v)$ represents the selection probability of antibody v, $fit(v)$ is the fitness value of number v antibody, $Maxfit(v)$ is the max fitness of the antibody, and c_v is the antibody v's concentration.

Select M optimal individuals according to the concentration of antibody from high to low to form a new population $X(t+1)$, thus the proportion of optimal individuals in the population can be increased to improve the local search ability.

Step 7: Self-adaptive crossover and mutation operation, self-adaptively select the crossover and mutation probability according to the diversity of population. Implement crossover towards M individuals based on the probability p_c to form the population $X_2(t)$, then respectively implement mutation towards M individuals in the population $X_2(t)$ based on the probability p_m to form population $X_3(t)$.

Step 8: Termination test, if termination criteria has been met, output the global optimal solution so far, otherwise, return to Step 3.

5. Numerical Optimization Experiment and Analysis

To test the effectiveness of the algorithm proposed in this paper in terms of the solution of to solve Multipeak optimization problem, the typical Multipeak function is optimized. See following formula for specific description of test function:

$$f = e^{-2\times ln2\times((x-0.08)/0.854)^2} \times \sin^6(5\pi(x^{3/4}-0.05)), x \in [0,1] \tag{6}$$

Function f has five peak values with non-uniform distribution and non-uniform altitude. Optimal values of these functions are difficult to search with genetic algorithm.

Antigens in the experiment are several peak values of the function to be solved, and the antibody is the independent variable x of the obtained peak value, and x is represented by the binary string with the length of 15. The parameter setting of two kinds of algorithm are shown in Table 1. The experimental computational results are shown in Figure 5 and 6.

Table 1. Parameters of Two Algorithm

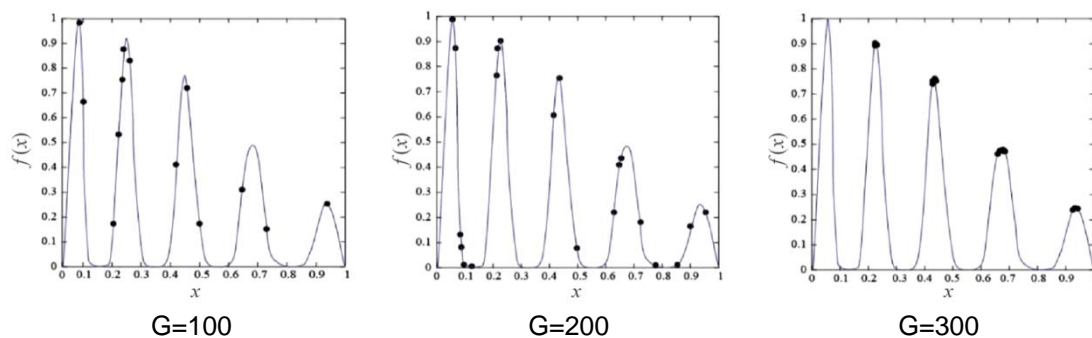| Algorithm	Crossover probability p_c	Mutation probability p_m	Concentration threshold value ε	Group size N	Memory cell length
Genetic algorithm	0.07	0.02		50	
SMIGA	0.75	0.15	0.85	50	Variable

G=100   G=200   G=300

Figure 5. Results of genetic algorithm

From above Figure 5 and Figure 6, we can see that, this SMIGA algorithm can still maintain convergence result in the case that the evolution generation is gradually increased,

and elements in the memory warehouse eventually converges to each peak value. The reason lies in that in the process of search, SMIGA can change the length of the memory cells and store those new antibodies non-similar to its original antibodies, thus making such algorithm more effectively find total extreme values. Based on the evaluation and analysis in such terms as the function convergence speed, ability to search the extreme value and the robustness, with better robustness, the algorithm performance in this paper shows larger advantage to better find each peak value of Multipeak function, thus can be successfully applied to Multipeak function optimization problems.
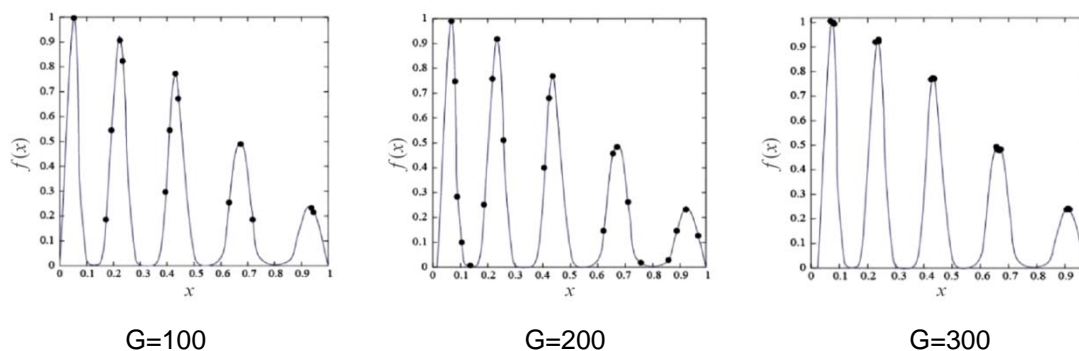

G=100   G=200   G=300

Figure 6. Results of SMIGA algorithm

6. Conclusion

Nowadays, intelligence algorithms have been used more and more extensively in numerous fields of scientific research and engineering practices and an increasing number of people begin to pay attention to them. By referring to the immune maturation mechanism of biology immune system and integrating genetic algorithm, this paper has proposed an self-adaptive multi-peak immune genetic algorithm, which significantly enhances the global convergence of the algorithm and the accuracy of the global extremum by adjusting the crossover and mutation probability and dynamically generating vaccine. The performance test function has shown that this algorithm has achieved better effects in solving multi-peak function optimization problems.

Acknowledgements

References

[1] Xuesong Yan, Qinghua Wu, Can Zhang, et al. An Improved Genetic Algorithm and Its Application. *TELKOMNIKA Indonesian Journal of Electrical Engineering.* 2012; 10(5): 1081-1086.
[2] Mlungisi Duma, Tshilidzi Marwala, Bhekisipho Twala, Fulufhelo Nelwamondo. Partial Imputation of Unseen Records to Improve Classification using a Hybrid Multi-layered Artificial Immune System and Genetic Algorithm. *Applied Soft Computing.* 2013; 13(12): 4461-4480.
[3] Soedibyo Soedibyo, Heri Suryoatmojo, Imam Robandi, et al. Optimal Design of Hydrogen Based Wind Microhydro using Genetic Algorithm. *TELKOMNIKA Indonesian Journal of Electrical Engineering.* 2012; 10(6): 1309-1318.
[4] VS Aragón, SC Esquivel, CA Coello. An Immune Algorithm with Power Redistribution for Solving Economic Dispatch Problems. *Information Sciences.* 2015; 295(20): 609-632.
[5] Ketan Dinkar Sarode, V Ravi Kumar, BD Kulkarni. Embedded Multiple Shooting Methodology in a Genetic Algorithm Framework for Parameter Estimation and State Identification of Complex Systems. *Chemical Engineering Science.* 2015; 134(29): 605-618.
[6] Karthik Balasubramanian, Basil Jacob, K Priya, K Sangeetha, N Rajasekar, et al. Critical Evaluation of Genetic Algorithm Based Fuel Cell Parameter Extraction. *Energy Procedia.* 2015; 75(8): 1975-1982.
[7] P Victer Paul, N Moganarangan, S. Sampath Kumar, et al. Performance Analysis over Population Seeding Techniques of the Permutation-Coded Genetic Algorithm: An empirical Study Based on Traveling Salesman Problems. *Applied Soft Computing.* 2015; 32(7): 383-402.

[8] Mahmoud M El-Sherbiny, Rashid M Alhamali. A Hybrid Particle Swarm Algorithm with Artificial Immune Learning for Solving the Fixed Charge Transportation Problem. *Computers & Industrial Engineering*. 2013; 64(2): 610-620.

[9] Fengqiang Zhao, Guangqiang Li, Chao Yang, Ajith Abraham, Hongbo Liu. A Human–computer Cooperative Particle Swarm Optimization Based Immune Algorithm for Layout Design. *Neuro Computing*. 2014; 132(20): 68-78.

[10] Poonam Savsani, RL Jhala, Vimal Savsani. Effect of hybridizing Biogeography-Based Optimization (BBO) Technique with Artificial Immune Algorithm (AIA) and Ant Colony Optimization (ACO). *Applied Soft Computing*. 2014; 21(8): 542-553.

[11] Seyed Abbas Taher, Muhammad Karim Amooshahi. New approach for Optimal UPFC Placement Using Hybrid Immune Algorithm in Electric Power Systems. *International Journal of Electrical Power & Energy Systems*. 2012; 43(1): 899-909.