

Multi-Population Genetic Algorithm Based on Adaptive Learning Mechanism



Jiawen Pan, Qian Qian, Yong Feng, and Yunfa Fu

Abstract Traditional genetic algorithm has some disadvantages, such as slow convergence, unstable, and easy to fall into local extreme. In order to overcome these disadvantages, an improved genetic algorithm is proposed in the present study. First, based on the analysis of advantages and disadvantages of learning mechanisms in literature, new improvements of learning mechanisms under the multi-population parallel GA are made. In previous studies, gene patterns from which other individuals can learn will be extracted from the excellent individuals of the population, this study improved the learning mechanism by adaptively changing the related control parameters, and dynamically controlling the process of the learning mechanism. Simulation results show that the new algorithm has a great improvement in many aspects of the global optimization, such as convergence speed, the accuracy of the solution, and stability.

Keywords Genetic algorithm · Adaptive · Multi-population parallel mechanism · Learning mechanism

1 Introduction

Genetic algorithm (GA) [1] is a global random search algorithm proposed in the early 1970s, which draws on the natural selection ideas and natural genetic mechanism of the biological world. GA has been widely used in Machine Learning, Control, Optimization, and other fields [2, 3].

J. Pan · Q. Qian (✉) · Y. Feng · Y. Fu

Faculty of Information Engineering and Automation, Kunming University of Science and Technology, Kunming 650500, Yunnan, China
e-mail: qianqian_yn@126.com

J. Pan · Q. Qian · Y. Feng

Yunnan Key Laboratory of Computer Technology Applications, Kunming University of Science and Technology, Kunming 650500, Yunnan, China

GA has great parallel optimization ability due to it can search multiple points by using fitness function as the search criterion. However, the algorithm also has defects such as “premature” convergence and slow convergence speed. Many methods have been proposed to solve these problems. It is an effective method to introduce learning mechanisms into GA. Hinton et al. [4] are the first researchers to introduce learning mechanisms to guide the evolution of GA algorithms. Candidates of the learning mechanisms are the optimization algorithms with strong local search capabilities, such as simplex method [5], hill-climbing method [6], etc. Literature [7] pointed out that most learning methods can be summarized as Lamarckian learning mechanism, Baldwinian learning mechanism, and the combination of these two to form a new learning mechanism. Literature [8] proposed a model learning mechanism under the guidance of the Lamarckian learning mechanism and the pattern theorem [1], but this mechanism uses a linear method to control related parameters, which may lead to the invalidity of the learning mechanism during operation. Based on previous studies, the present paper proposes Multi-population Genetic Algorithm based on Adaptive Learning Mechanism (ALMGA), which can dynamically control the calculation process of the learning mechanism under a multi-population GA according to the different conditions of the individual fitness during the evolution. Finally, simulation results show that ALMGA has great improvements in global optimization, local search, and convergence speed.

2 Almga

2.1 Multi-Population Parallel Mechanism

Multi-population parallel genetic algorithm is a good method to improve the performance of genetic algorithm [9], this mechanism can be divided into three basic types: master-slave model, fine-grained model, and coarse-grained model.

In this paper, the parallel mechanism based on coarse-grained model is introduced to perform independent evolution of multiple populations. Each population is configured with different parameters and initial values, and data migration is performed regularly. Specifically, at the beginning of each evolutionary generation, the population exchanges the optimal individual and gene pattern with the public area (see Sect. 1.2 for details).

2.2 Basic Concepts of Learning Mechanism

Definition 1 Let $H = \{H_1, H_2, \dots, H_m\}$ be the collection of all populations, M is the number of populations, N is the number of chromosomes in each population.

Definition 2 Age and overage. For any population H_m and $\forall x_i \in H_m$, let $x_{ig.age}$ represents the age of the chromosome x_i in g generation. For chromosome x_i , its initial age is 0, the age update formula (1) is as follows:

$$x_{ig+1.age} = x_{ig.age} + \alpha \left(\frac{H_{mg.ave}}{x_{i.fit}} \right) \quad (1)$$

Among them, $i = 1 \sim N$, α represents a control parameter. $H_{mg.ave}$ represents the average fitness of the population H_m in g generation, $x_{i.fit}$ represents the fitness of the chromosome x_i . If the fitness of the chromosome is lower than the average fitness of the population for a long time, its age will be high, which means the “evolution potential” of the chromosome is worse. When the age exceeds a pre-defined age value, the chromosome becomes an overage chromosome and will be eliminated in subsequent evolution. $H_{m.old}$ represents the number of overage individuals in population H_m .

Definition 3 Excellent rate. For any population H_m , H_e represents the excellent rate of H_m (see 1.4 for the calculation formula), the number of excellent individuals in H_m is $r = H_e * N$, the r individuals with highest fitness in H_m are called excellent individuals.

Definition 4 Gene pattern and pattern extraction. For any population H_m , let any two chromosomes of the population as $x = \{x_1, x_2, \dots, x_j\}$ and $y = \{y_1, y_2, \dots, y_j\}$, $x_j, y_j \in \{0, 1\}$. The chromosome generated by the above two chromosomes through

the formula $z = x \times y = \left\{ \begin{array}{l} x_j, x_j = y_j \\ *, x_j \neq y_j \end{array} \right\}$ are called gene pattern, $z_j \in \{0, 1, *\}$, “*”

represents undetermined genes. In the present algorithm, the gene pattern is obtained by extracting all excellent individuals of the population in the current generation, $H_{mg.scheme}$ represents the gene pattern of the population H_m in the g generation.

Definition 5 pattern learning. For any population H_m , let any chromosome of the population as $x = \{x_1, x_2, \dots, x_j\}$, the chromosome and gene pattern $H_{mg.scheme} = \{z_1, z_2, \dots, z_j\}$ perform pattern learning behavior through formula:

$x \odot H_{mg.scheme} = \left\{ \begin{array}{l} x_j, x_j = z_j \text{ 或 } z_j = * \\ z_j, x_j \neq z_j \text{ 且 } z_j \neq * \end{array} \right\}$ get a new chromosome x_{new} . For example,

when $x = 1010110$ and the gene pattern $H_{mg.scheme} = 1*0100*$, then after learning $x_{new} = 1001000$.

Definition 6 The weight of gene pattern. For any population H_m , $H_{mg.weight}$ represents the weight of $H_{mg.scheme}$, the higher the weight, the more “excellent” $H_{mg.scheme}$ is, the lower the weight, the more “inferior” $H_{mg.scheme}$ is (see Sect. 1.7 for the calculation formula).

Definition 7 Best gene pattern. For any population H_m , $H_{m.bscheme}$ represents the most “excellent” (with highest weight) gene pattern of the population until current generation.

Definition 8 Global best gene pattern. During the evolution of the algorithm, the best gene pattern (with highest weight) among all populations is called the global best gene pattern, which is represented by $H_{bscheme}$.

Definition 9 Common areas. Multi-population parallel GA requires regular data migration and exchange. In the present algorithm, a common area is used to receive the gene patterns and optimize the chromosomes shared by all populations, the common area will compare the weights of different gene patterns to select $H_{bscheme}$ and compare the fitness values to select the global optimal individual. Common area constantly monitors all populations and transfers $H_{bscheme}$ to the population that requests learning between populations.

2.3 Operation Process of Learning Mechanism

Learning mechanism includes two learning method: learning within populations (LWP) and learning between populations (LBP), which are controlled by possibilities P_i and P_o , respectively.

The process of LWP is as follows:

- (a) Calculate H_e of the population;
- (b) Select r chromosomes as excellent individuals of the population according to the fitness values;
- (c) $H_{mg.scheme}$ is obtained by pattern extraction from the excellent individuals, compare $H_{mg.scheme}$ with $H_{m.bscheme}$ of the population and update $H_{m.bscheme}$;
- (d) P_i controls chromosomes of the population to conduct pattern learning based on $H_{m.bscheme}$, and update fitness values and age of chromosomes.
- (e) If it reaches the required generation of information migration, transfer $H_{m.bscheme}$ and the best individual in the common area.
- (f) The common area compares the $H_{m.bscheme}$ and optimal chromosomes passed from all populations with the existing $H_{bscheme}$ and global optimal individuals, and updates the $H_{bscheme}$ and global optimal individuals.

The process of LBP is as follows:

- (a) P_o controls whether the population to conduct LBP and with a positive result, LBP sends a learning request to the common area;
- (b) Common area sends $H_{bscheme}$ to the above population;
- (c) The population first deletes overage individuals, then generates new chromosomes to learn from $H_{bscheme}$ and to fill the vacancy of eliminated overage individuals, finally update fitness values and age of chromosomes.

In LWP, individuals can improve their own traits in less generations by learning the common genes of excellent individuals, by this way, the evolution directions of the individuals can be guided. In contrast, LBP guides the evolution of the population, a new chromosome is produced to learn from $H_{bscheme}$ and replace overage individuals, thus improving the “evolution potential” of population by improving the diversity of the population. Together, the LWP and the LBP make full use of their own advantages and improve the performance of the algorithm.

2.4 Adaptively Change the Value of the Excellent Rate

Excellent rate H_e selects the excellent individuals of the population at each iteration, and guides the evolution direction of the population through the gene pattern obtained from these excellent individuals. Paper [8] uses a linear formula to calculate H_e , the value of H_e will increase linearly when the average fitness of the population increases with the number of iterations. A high value means that there are more excellent individuals. Therefore, the gene model obtained by pattern extraction will be full of non-deterministic gene, making the gene pattern invalid to guide the evolution. In order to avoid the condition that the gene patterns are full of undetermined genes, this paper proposes a calculation formula (2) to adaptively control the excellent rate values based on evolutionary generations:

$$H_e = \begin{cases} H_{e1} + \frac{H_{e2}-H_{e1}}{2} \sin\left(\frac{g}{G}\pi\right) & g \geq \frac{G}{2} \\ H_{e1} + \frac{H_{e2}-H_{e1}}{2} (1 - \sin\left(\frac{g}{G}\pi\right)) & g < \frac{G}{2} \end{cases} \quad (2)$$

In the formula, (2): H_{e1} is the minimum excellent rate; H_{e2} is the maximum excellent rate; g is the current evolution generation number; G is the maximum evolution generation number of the algorithm.

In the Fig. 1, the dotted line represents the variation of H_e in the paper [8], which follows a linear transformation method, the solid line indicates the change curve of the H_e following current formula (2). As we can see, H_e has a greater value than linear transformation at the beginning of the algorithm. This is because in the early stage of evolution, the average fitness of the population is relatively low, and excellent individuals contain more inferior genes, so formula (2) adaptively increases the value of H_e to increase the number of excellent individuals. As a result, the pattern extraction based on a large number of excellent individuals will reduce the probability of the inferior genes entering the gene pattern. During the late stage of the algorithm, the average fitness of the population and the similarity between the genes of the excellent individuals are relatively high, the value of H_e is adaptively reduced to ensure that only the truly excellent genes can be extracted into the gene pattern. In addition, adaptively change the value of the excellent rate can also avoid the phenomenon that the gene pattern in the middle and late stages of the algorithm is full of undetermined genes so that the gene pattern can better guide the evolution direction of the population.

2.5 Adaptively Change the Value of P_i

Learning within populations (LWP) relies on P_i to control the chromosomes of the population to learn $H_{m.bscheme}$. Paper [8] uses a formula that follows a linear transformation to calculate P_i . Such formula does not fully consider the evolution situation of the population. Since the learning frequency gradually increases along with the increase of the average fitness of the population, the population may fall into “premature” convergence. In the present algorithm, both the generations and the evolution situations (i.e., fitness values) of the population are considered. In the early stage of the algorithm, the quality of individuals in the population is generally poor, so the guidance to the population should be strengthened by increasing the value of P_i . As the evolution continues, P_i should be gradually reduced to maintain the diversity of the population, so as to increase the converging probability to the best solution. In addition, the population with low fitness values should take a large value of P_i to strengthen the guidance, and the population with high fitness values should reduce P_i , so as to ensure the diversity of the population, avoiding the phenomenon of “premature” convergence. Based on the above reasons, this study designs a formula for calculating P_i as follows:

$$P_i = P_{i1} + (P_{i2} - P_{i1}) \frac{1}{1 + \exp(\frac{g}{G} - 1)} \left(1 - \left(\frac{H_{m.ave}}{H_{m.best}} \right) \right) \quad (3)$$

In formula (3): P_{i1} is the minimum probability of LWP; P_{i2} is the maximum probability of LWP; g is the current evolution generation; G is the maximum

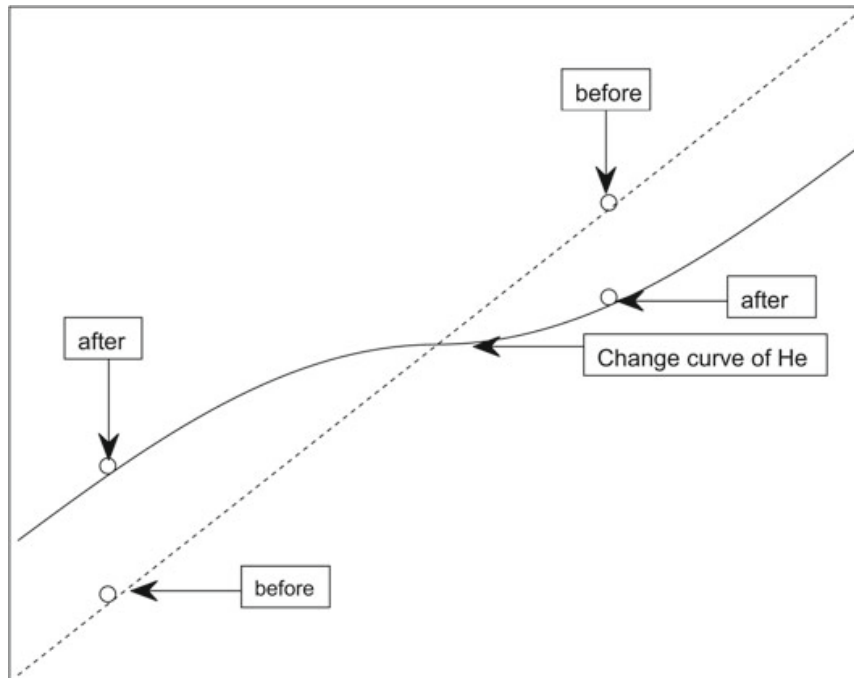


Fig. 1 Adaptive curve of excellent rate

evolution generations; $H_{m.ave}$ is the average fitness of the population; $H_{m.best}$ is the maximum fitness of the population.

2.6 Adaptively Change the Value of P_o

Learning between populations (LBP) controlled by P_o . Paper [8] does not fully consider the influence of overage individuals, so that there may be a situation in which too many overage individuals are eliminated, resulting in instability of the algorithm, or there may be no overage individuals at all, thus making learning invalid. In all, when there are many overage individuals or when the population has a low fitness value, P_o should be increased. Based on the above reasons, formula (4) is used to adaptively adjust the probability P_o of LBP according to the number of overage individuals in the population and the evolution situations of the population:

$$P_o = 4\sqrt[3]{\frac{H_{m.old}}{N}}(P_{o1} + (P_{o2} - P_{o1}))\cos\left(1 - \left(\frac{\max(H_{m.ave}) - H_{m.ave}}{\max(H_{m.ave}) - \min(H_{m.ave})}\right)\right) \quad (4)$$

In the formula (4): P_{o1} is the minimum probability of LBP; P_{o2} is the maximum probability of LBP; g is the current evolution generation; G is the maximum evolution generation; $H_{m.old}$ is the number of overage individuals in the population; N represents the total number of chromosomes in the population; $\max(H_{m.ave})$ is the maximum average fitness value of all populations; $H_{m.ave}$ is the average fitness of the population; $\min(H_{m.ave})$ is the minimum average fitness value of all populations.

According to formula (4), the number of overage individuals in the population determines the value of P_o . As the evolution continues, the number of overage individuals in the population continues to increase and the P_o value also gradually increases. Therefore, every time LBP is performed, the number of overage individuals in the population will be always maintained in a reasonable number, thus avoiding the situations that there are no overage individuals or too many overage individuals in the population. Finally, a cosine function is introduced to nonlinearly control the learning probability of populations, which can make the learning smooth and maintain the stability of the algorithm. Specifically, for populations whose average fitness value at the range of $[\min(H_{m.ave}), (\max(H_{m.ave}) - \min(H_{m.ave}))/2]$, the population should catch up with other populations, so it has a larger P_o value; for populations whose average fitness value at the range of $[(\max(H_{m.ave}) - \min(H_{m.ave}))/2, (\max(H_{m.ave}) + \min(H_{m.ave}))/2]$, the quality of the genes is good, so the P_o value of the population is slightly increased to get better learning results; in other conditions, a small P_o value is assigned, so as to emphasize the learning within populations. By nonlinearly changing the learning probability of different

populations, LBP can better play the role of guiding the evolution direction and strengthening the local search ability of the population.

2.7 The Weight of the Gene Pattern

The learning mechanism can guide the evolution direction of the population, but it is not ensured that the gene pattern of current generation is better than the gene pattern of previous generation. Therefore, an evaluation function is designed in the present algorithm to calculate the weight of the gene pattern, so that the algorithm can use the weight to compare the quality of different gene patterns. The calculation formula is as follows:

$$H_{mg.weight} = \bar{f} \left(1 - \frac{3G - 2\sqrt{g}}{5G} \right) + H_{m.best} \frac{3G - 2\sqrt{g}}{5G} \quad (5)$$

In the formula (5): $H_{m.best}$ is the fitness value of the best individual in the population; \bar{f} is the average fitness value of the remaining excellent individuals within the population except $H_{m.best}$; g is the current evolution generation; G is the maximum evolution generations.

In the early stage of evolution, the average fitness of the population is relatively low, most chromosomes in the population contain “inferior” genes. Under such conditions, the ability of the best individuals in the population to guide the evolution direction should be strengthened, and the impact of other excellent individuals should be weakened. As the algorithm continues, the average fitness of the population gradually increases, and the similarity between the best individuals and other excellent individuals also increases, in order to avoid the misguided evolution direction by the local optimal solution, the impact of $H_{m.best}$ should be reduced and the impact of \bar{f} should be increased.

2.8 The Flow of ALMGA

The algorithm execution process is as follows:

Set the control parameters of the algorithm, including population number, chromosome length, age, etc., initialize the population and public area.

Calculate the fitness of individuals in the population.

All populations independently evolution according to their own evolution parameters, perform genetic operations and update individual fitness and age.

All populations calculate the number of excellent individuals according to formula (2) and perform pattern extraction, update $H_{m.bscheme}$, and pass the optimal

individual and $H_{m.bscheme}$ to the common area. The common area receives data from various populations and updates the global optimal individual and $H_{bscheme}$.

Individuals in the population determine whether to perform LWP by Eq. (3).

Each population determines whether to perform LBP by Eq. (4).

Check the termination condition, and go to (8) if it meets, otherwise go to (2).

The optimal solution is outputted.

3 Simulation Test

In order to verify the performance of the ALMGA algorithm proposed in this article, this section conducts simulation tests based on four standard test functions given in Table 1. The HGA proposed in paper [6], the PGABL proposed in paper [8] and the ALMGA algorithm proposed in this paper are compared.

Both HGA and ALMGA use the same parameters, and the parameters of PGABL are consistent with that in the paper [8]. The above algorithms all use binary coding, fitness proportion selection, single-point crossover, and multi-point mutation. The control parameters of the algorithm are as follows: the population number is 4, the chromosome size of each population is 50, the chromosome length is $D * 40$, and the number of algorithm generations is 400. In HGA, $p_{m1} = 0.2$, $p_{c1} = 0.8$, $p_{m2} = 0.05$, $p_{c2} = 0.5$, $p_{m3} = 0.1$, $p_{c3} = 0.6$. In PGABL $a = 1$, $life = 15$, $P_c = 0.8$, $P_m = 0.06$. In ALMGA, $\alpha = 1$, $life = 15$, $P_{m1} = 0.2$, $P_{c1} = 0.8$, $P_{m2} = 0.05$, $P_{c2} = 0.5$, $P_{m3} = 0.1$, $P_{c3} = 0.6$, $P_{e1} = 0.1$, $P_{e2} = 0.2$, $P_{i1} = 0.3$, $P_{i2} = 0.5$, $P_{o1} = 0.1$, $P_{o2} = 0.15$.

Table 2 shows the test results after each of the three algorithms are executed 30 times. It can be seen from the test results that compared with HGA and PGABL, the optimal value of the function solved by ALMGA is closer to the theoretical optimal value of the function, which indicates the algorithm has high accuracy. The number of convergences of ALMGA is close to the total number of experiments, among them, the number of convergences for solving functions f_1 and f_2 is consistent with the total number of experiments, which indicating the algorithm has good stability. It can be seen from the result of PGABL to solve function f_1 that although the number of convergence is low, but the average convergence iterations is also low, which indicating the learning mechanism has given the algorithm better optimization capabilities. The PGABL to solve function f_2 has less convergence generation to meet convergence accuracy, this is because the global optimal value of f_2 is surrounded by the local optimal value. PGABL is easy to fall into the local optimal value during the algorithm optimization process, it shows that the algorithm has poor ability to jump out of the local optimal value, and also shows that ALMGA has a better ability to jump out of the local optimal value. There are two extreme cases when solving $f_3 - f_4$, PGABL solves $f_3 - f_4$ without convergence, while HGA has fewer times of convergence, and ALMGA has more times of convergence, which indicating ALMGA has better global optimization and convergence capabilities.

Table 1 Test functions

	Function name	Function expression	Domain	Optimal value
f_1	<i>Camle</i>	$\left(4 - 2.1X_i^2 + \frac{X_i^4}{3}\right)X_i^2 + X_iX_{i+1} + (-4 + 4X_{i+1}^2)X_{i+1}^2$	$[-10, 10]$	-1.031628
f_2	<i>haystack</i>	$\left[\frac{3}{0.05 + (X_i^2 + X_{i+1}^2)}\right]^2 + (X_i^2 + X_i^2)^2$	$[-5.12, 5.12]$	3600
f_3	<i>Rastigrin</i>	$\sum_{i=1}^D (X_i^2 - 10 \cos 2\pi X_i + 10)$	$[-10, 10]$	0
f_4	<i>Ackley</i>	$-20 \exp\left(-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D X_i^2}\right) - \exp\left[\frac{1}{D} \sum_{i=1}^D \cos 2\pi X_i\right] + 20 + e$	$[-2.048, 2.048]$	0

Table 2 Comparison results of function test

	Accuracy	Algorithm	Optimal value	Best convergence iterations	Average convergence iterations	Convergence times	Convergence rate
f_1	10^{-5}	HGA	-1.03162470255	131	181	20	0.667
		PGABL	-1.03162615926	114	163	13	0.433
		ALMGA	-1.03162845151	24	61	30	1.000
f_2	10^{-3}	HGA	3599.999004362	166	258	14	0.467
		PGABL	3599.999216859	131	155	2	0.067
		ALMGA	3599.999993133	41	72	30	1.000
f_3 (D = 10)	10^1	HGA	6.057019493037	274	340	7	0.233
		PGABL	25.86890531471			0	0
		ALMGA	2.484773167789	149	268	26	0.867
f_4 (D = 25)	10^0	HGA	0.753118330563	292	331	3	0.133
		PGABL	2.018254436453			0	0
		ALMGA	0.401447729784	137	192	28	0.933

4 Conclusion

In order to overcome the shortcomings of traditional genetic algorithms, the Multi-population Genetic Algorithm Based on Adaptive Learning Mechanism is proposed in the present study. The algorithm introduces multi-population parallel mechanisms to strengthen the global optimization capabilities of genetic algorithms, and introduces learning mechanisms to guide the evolution, and strengthens local search capabilities of the algorithm. At the same time, it makes adaptive improvements to the learning mechanism, so that the algorithm can dynamically control the process of the learning mechanism. Finally, through simulation experiments on four standard test functions, the results show that the ALMGA proposed in this paper significantly improved the searching ability and convergence ability of the algorithm.

References

1. Holland HJ (1975) Adaptation in natural and artificial systems. Ann Arbor Michigan Univ Press 6(2):126–137
2. Zhou Y, Zhou L, Wang Y et al (2017) Application of multiple-population genetic algorithm in optimizing the train-set circulation plan problem. Complexity
3. Li L, Tang Y, Liu J et al (2013) Application of the multiple population genetic algorithm in optimum design of air-core permanent magnet linear synchronous motors. Proc CSEE 33 (15):69–77
4. Hinton EG, Nowlan JS (1987) How learning can guide evolution. Complex Syst 1(43):495–502
5. Tang W, Chen Y, Zhang M (2019) An energy balanced routing algorithm with simplex method. Comput Technol Dev 29(3):55–59
6. Zheng M, Zhuo M, Zhang S et al (2017) Reconstruction for gene regulatory network based on hybrid parallel genetic algorithm and threshold value method. J Jilin Univ Eng Technol Ed 47 (2):624–631
7. Zhibo L, Qitao H, Hongzhou J (2009) Mixed application of two learning mechanisms in genetic algorithm. Syst Eng Electron 31(8):1985–1989
8. Zhang G, Wu Z, Liu X (2005) Parallel genetic algorithm based on learning mechanism. Comput Appl 25(2):374–376
9. He W, Wang J, Hu L (2009) The improvement and application of real-coded multiple-population genetic algorithm. Chin J Geophys 52(10):2644–2651