



A self-adaptive genetic algorithm with improved mutation mode based on measurement of population diversity

Na Sun¹ · Yong Lu¹

Received: 8 January 2018 / Accepted: 16 March 2018 / Published online: 5 April 2018
© The Natural Computing Applications Forum 2018

Abstract

Genetic algorithm (GA) is an important and effective method to solve the optimization problem, which has been widely used in most practical applications. However, the premature convergence of GA has unexpected effect on the algorithm's performance, the main reason is that the evolution of outstanding individuals multiply rapidly will lead to premature loss of population's diversity. To solve the above problem, a method to qualify the population diversity and similarity between adjacent generations is proposed. Then, according to the evaluation of population diversity and the fitness of individual, the adaptive adjustment of crossover and mutation probability is realized. The results of several benchmark functions show that the proposed algorithm can search the optimal solution of almost all benchmark functions and effectively maintain the diversity of the population. Compared with the existing algorithms, it has greatly improved the convergence speed and the global optimal solution.

Keywords Genetic algorithm · Mutation mode · Population diversity · Self-adaptive genetic algorithm

1 Introduction

The optimization theory has been studied and developed in many aspects, and it is widely used in the fields of industrial, agricultural, economic and military, etc. Until the advent of modern computers, many optimization methods that require an enormous amount of computation can be successfully calculated, and it make the optimization theory and method be greatly improved [1]. After a century of development, the theory of optimization has made great achievements. However, there are still some basic problems that need to be resolved. Genetic algorithm was first proposed by Prof. Holland in 1975, which has the characteristics of fast search, strong randomness, simple process and strong flexibility [2]. Absolutely, it is very suitable for applying in the fields of function optimization and combination optimization [3, 4].

The theoretical studies of genetic algorithms include the precocious convergence of the algorithm, the ability to search, the setting of control parameters, and the corresponding improvements, and the research work can be launched from the point of views of theory or experiments [5, 6]. Due to some subjective and objective factors, it is impossible for researchers to master all useful information in all aspects of the problem, which make the effect of different algorithms with unexpected results [7]. Moreover, the impact of the key problems on the performance of the algorithm is not very clear, such as the search performance of genetic operators, the coordination ability among different genetic operators, the speed of convergence of genetic algorithm and the complexity of algorithm [8–10]. Therefore, the above problems should be further discussed, and the better design of the genetic algorithm can be presented to make the desired performance gradually.

At present, the research of GA is relatively mature and extensive, but there are plenty of problems in the application method and theory of the genetic algorithm needs to be resolved. For enhancing the performance of the algorithm, researchers have tried various methods to improve the population diversity, crossover and mutation of GA [11, 12]. Furthermore, most of researches focus on the detailed aspects, i.e. setting of control parameters of

✉ Yong Lu
yong@muc.edu.cn

Na Sun
2012048@muc.edu.cn

¹ School of Information Engineering, Minzu University of China, Beijing 100081, China

genetic algorithm, the types of problems to be dealt with, the coding methods of feasible solutions, the selection of selection methods, the convergence analysis of algorithms and the measurement of the performance of algorithms [13–15]. Inevitably, the completeness and incompleteness are inherent in the theory of models. Therefore, the following research work should make the existing model theory perfect to form a more systematic and rationalized model theory. Further, in the light of the shortcomings of the existing model theory, other methods for simulating biological evolution are sought on the basis of bionics.

The rest of this paper is organized as follows. Section 2 discusses related work, followed by the problem description and analysis in Sect. 3. The improved mutation mode based on measurement of population diversity is discussed in Sect. 4. Section 5 shows the simulation experimental results, and Sect. 6 concludes the paper with summary and future research directions.

2 Related work

In recent years, the research of genetic algorithms has been paid much attention by many scholars and has made remarkable achievements. Chen et al. [16] proposed the ethnic group evolution algorithm, which uses the ethnic group mechanism to design an ethnic group classification method for binary coding. Meanwhile, based on the structure of the ethnic group, the dual evolution algorithm and the corresponding ethnic group operators are formed, which can significantly improve the population's precocious ability and search efficiency. In [17], a good point set-based genetic algorithm (GGA) is proposed, in which the crossover operation of GA is redesigned with the best point set principle in number theory. From the experiment results, the evolutionary speed of the algorithm is improved, but there is still a lack of accuracy. In [18], referring to the biological model of bee colony breeding, a bee evolutionary genetic algorithm (BEGA) is introduced. The best individuals in the population are intersecting with other individuals, and the algorithm is superior to other GA in both the success rate and the convergence speed. However, the algorithm has some defects in theory. Its crossover process is to produce the best individuals and other individuals in the past dynasties, and it will cause the algorithm to converge to the local optimal solution to some extent.

Besides, many researchers have proposed different improvements in combination with the idea of elites and coevolution. In [19], a hybrid self-adaptive orthogonal genetic algorithm (HSOGA) based on orthogonal experimental design method is proposed to resolve the global optimization problems. To maintain the diversity of

generations, a self-adaptive method for the selection of orthogonal crossover operators based on the orthogonal experimental design is proposed, which can divide the parents into several sub-vectors with respect to the parents' similarity and produce a small but representative set of points as the potential offspring. In [20], a hybrid competitive coevolutionary genetic algorithm is introduced to resolve the Rosenbrock function optimization problem. The deficiencies are that the algorithm uses the strategy of dividing the solution space, and it is prone to fall into the local minimum. In [21], a novel parallel genetic algorithm is proposed, which divides the whole population into sub-populations to be evolved, to solve the multidimensional knapsack problem. Due to same evolutionary strategy be applied in all populations, the evolutionary process is relatively independent and the exploration ability of the algorithm will be reduced to some extent. Moreover, the parameters are set manually to cause the fluctuation of algorithm's performance. In [22], a double elite coevolutionary genetic algorithm (DECGA) is introduced, which generates two different elite individuals with high fitness individuals for the evolutionary operation. By utilizing the different evaluation functions, the members of two sub-populations can be chosen to balance the capability of exploration by means of different evolutionary strategies.

In a local range, the searching ability of genetic algorithms can be reduced. The main reason is that the emergence of this precocious convergence makes the role of the crossover operators and mutation operators of the genetic algorithms weaken. Therefore, even after many generations of evolution, there will not be an individual with greater fitness in the population. In order to solve the problem of premature convergence of GA effectively, researchers have proposed many improved GA algorithms. In [23], the elitist version of the non-homogeneous genetic algorithm is proposed to change the mutation and crossover probabilities during the entire phase of the algorithm's evolution, and the sure convergence to optimal populations is proven. Also, an approach to weak ergodicity of a non-homogeneous Markov chains is proposed to search for optimal points of functions, which adopts unconventional methods without Dobrushin's ergodicity coefficient [24]. In [25], a population size reduction methods for a typical GA optimization are presented, which aim to increase efficiency and accuracy of the traditional algorithms by using Monte Carlo simulations. In [26], adaptive regulation scheme of mutation rate is discussed to improve the convergence of the genetic algorithm, and autoregressive integrated moving average model (ARIMA) selection and maximum likelihood estimation are introduced to keep the mutation probability at a lower level, instead of staying in the high range for a long time. In [27], a fuzzy genetic algorithm is proposed for solving binary-encoded combinatorial

optimization problems, most of which are dependent on the genetic operators and the type of crossover operator. Also, a novel mechanism for the selection of crossover operator and probability based on fuzzy logic controller is proposed in according with the population diversity.

3 Problem description and analysis

In the process of generation, the fitness function should be defined to evaluate the chromosomes, which is related to the objective function of the optimization problem, and the genetic operators will select chromosomes according to the evaluation values. Owing to use original evaluation value for search, the selection of fitness function is very important, which directly affects the speed of convergence of GA and determined whether to find the optimal solution. By employing the original evaluation value, the individuals with high fitness chromosomes in the population need much longer than the ones with low fitness and fitness scaling is an important element affecting the evolutionary performance of genetic algorithm. In addition, in general, the decrease in population diversity and the larger similarity between adjacent generations often appear in the later period of the algorithm. To alleviate the effect, the population diversity is regarded as an evaluation index of population's quality. However, when the range of variables is large, those methods will cause too much computation or lead to no obvious measurement effect. In addition, if the new individual is deficient, the new populations will be very similar to the original population. The similarity between adjacent generations is rather high, and it is necessary to vary the probabilities of crossover urgently, which can ensure the new generation with more differentiated population.

In general, crossover operation is the main way to produce better individuals in genetic algorithm. Assuming that the roulette selection method is adopted in GA and the individual fitness value is positive, the probability of being chosen for an individual with gene η at time t can be given as:

$$p(\eta, t) = M \frac{p(\eta, t-1)f(\eta)}{\sum f(H)} \quad (1)$$

where $p(\eta, t)$ denotes the probability that the gene η is selected at time t , M is the number of individuals in the population and $f(\eta)$ indicates the fitness value of gene η . H denotes the all populations and $\sum f(H)$ is the sum of fitness value of all populations at time t .

Suppose that $\hat{f}(t)$ is the average fitness value of all populations at time t , we have

$$f(\eta) - \hat{f}(t) = c_i \hat{f}(t). \quad (2)$$

Then, the probability of the gene η being selected at time t can be given as:

$$p(\eta, t) = p(\eta, t-1)(1 + \varphi_t). \quad (3)$$

Considering from the first occurrence of gene η in the selection operator, the total probability of the gene η being selected at time t can be obtained. Hence,

$$p(\eta, t) = \prod_i^t p(\eta, i)(1 + \varphi_n) \quad (4)$$

where $p(\eta, i)$ denotes the probability of the first occurrence of a gene η , and i denotes that the first time the gene η appears in the selection operator.

Then, the probability of gene η , which participates in the crossover operation to generate new individuals, can be expressed as:

$$p'(\eta, t) \leq pc(t) \frac{M[1 - \prod_i^t p(\eta, i)(1 + \varphi_n)]}{M - 1} \quad (5)$$

where $p'(\eta, t)$ denotes the probability of gene η being selected to be the parent for new offspring's generation. $pc(t)$ is the crossover probability at time t .

As can be seen from the derivation, the probability of producing new individuals by cross-operation is mainly related to the variables: $pc(t)$, $p(\eta, i)$ and c_n . Among them, $p(\eta, i)$ is a random variable and $pc(t)$ is a subjective value. For φ_n , it is related to the fitness value of the gene η and also uncontrollable.

Assume that the optimal individuals with size of η is satisfied with the condition $f(\eta) - \hat{f}(t) \geq 0$, i.e. $\varphi_t \geq 0$. Then the upper limit of $p'(\eta, t)$ will decrease with the increase in t in approximate exponent trend. The crossover operation will demonstrate insensitively. In fact, this phenomenon is particularly evident in the late phase of the evolution. Therefore, if $f(\eta)$ is not a global optimal fitness value, it is difficult to produce better individuals and the algorithm will result in falling into local convergence.

Therefore, the following conclusions can be obtained. In the early stage of evolution, new individuals are mainly produced by cross-operation, which could set larger crossover probability and smaller mutation probability. However, during the late stage of evolution, the probability of generating new individuals will be rather lower than in the early stage. Considering that mutation operation poses a greater impact on the production of new individuals, we can increase mutation probability appropriately and improve mutation operation to avoid falling into local optimal solution. Moreover, since that the generation of new individuals is related to the relative size of individual fitness values, we need to accurately evaluate the evolution process and the setting of genetic operators should be taken into account the impact of individual fitness and evolution process.

4 Improved mutation mode based on measurement of population diversity

4.1 Evaluation of population quality

The main reason for the genetic drift or early convergence in GA is due to the loss of population diversity during the evolutionary process. In order to maintain population diversity, researchers put forward many effective algorithms, such as eugenics, selection of effective place for crossover and mutation, and preservation of effective gene. However, most of those algorithms usually ignore the correlation between the genetic operation and the internal evolution state.

In order to solve the problem of population diversity reduction and the increment of similarity between adjacent generations, a method for determining the diversity of population and the similarity of adjacent generation population is introduced in this section. By utilizing the average, minimum and maximum fitness of individual fitness and the distribution of individual population comprehensively, the measure function of population diversity is designed as an evaluation index of population quality. To measure the similarity between adjacent generations, cosine similarity theory is introduced to define the similarity between the t th generation population and its offering. The similarity function can be defined as:

$$\delta(t) = \frac{\sum_{i=1}^N x_i^t x_i^{t-1}}{\sqrt{\sum_{i=1}^N (x_i^t)^2 + \exp(-10)} \sqrt{\sum_{i=1}^N (x_i^{t-1})^2 + \exp(-10)}} \times \prod_{i=1}^N \frac{S(x_i^t) \cap S(x_i^{t-1})}{S(x_i^t) \cup S(x_i^{t-1})} \quad (6)$$

where x_i^t denotes the actual value of i th individual of generation t . $S(x_i^t) \cap S(x_i^{t-1})$ denotes the size of the similar elements set, and $S(x_i^t) \cup S(x_i^{t-1})$ the size of the union for all elements from the sets $S(x_i^t)$ and $S(x_i^{t-1})$.

If the molecule of the formula (6) is equal to 0, the value of $\delta(t)$ will be set to 1. It ensures that the $\delta(t)$ will be close to 0 as the similarity between the adjacent two generation populations is low and $\delta(t)$ will approach to 1 conversely.

In addition, a measure function of population diversity is designed by using the average value, minimum and maximum value of individual fitness and the distribution of individual population, which can be defined as:

$$\gamma(t) = \frac{\sigma(t) - \min\{\sigma(t)\}}{\max\{\sigma(t)\}} \times \cos\left(\frac{\pi}{2} \times \left(1 - \frac{f_{\text{avg}}^t - f_{\text{min}}^t}{f_{\text{max}}^t - f_{\text{min}}^t}\right)\right) \quad (7)$$

where f_{avg}^t , f_{max}^t and f_{min}^t is the average value, the maximum and the minimum of the population fitness, respectively.

$\sigma(t)$ is the variance of the fitness of the generation t , and $\max\{\sigma(t)\}$ and $\min\{\sigma(t)\}$ denote the maximum and minimum value of the population variance at generation t , respectively. From the expression, the value of $\gamma(t)$ approximates to 1 means the higher population diversity. Otherwise, the closer the $\gamma(t)$ to 0, the lower the population diversity is.

With the increase in evolution generations, the trend of population diversity turns into decline gradually from 1 to 0. At the same time, the similarity between adjacent generations increased from 0 to 1. Therefore, we define the function $\phi(t)$ as the evaluation of population quality:

$$\phi(t) = \gamma(t)(1 - \delta(t)). \quad (8)$$

Obviously, when $\phi(t)$ is close to 1, the population diversity is much higher. On the contrary, when $\phi(t)$ approaches to 0, it indicates that the individuals of the populations tended to be the same.

4.2 Adjustment of operation probability

When the population has high diversity, the individuals in the population contain more different patterns, which indicate the better potential of searching for optimal individuals. At this point, the crossover probability should be appropriately increased and the mutation probability be reduced to ensure adequate mode reorganization among individuals. And this strategy can promote effective mode propagation in individuals and increase the breadth of search. Vice versa, when the population diversity is low, i.e., $\phi(t)$ is smaller, the optimal individuals in the population tend to be the same, and the probability of variation should be improved. In this case, the new individual must be introduced to break the state of better individual convergence and avoid the local optimum. Hence, the specific adjustment method of crossover operator and mutation operator is discussed as follows:

$$\begin{cases} pc_1(t) = pc_1 + \phi(t) \frac{(pc_2 - pc_1)}{2} \\ pc_2(t) = pc_2 \end{cases} \quad (9)$$

In which pc_1 and pc_2 denote the range of value of the initial crossover probability, and $pc_2 > pc_1$. $pc_1(t)$ and $pc_2(t)$ are the corresponding value of the crossover probability of generation t .

The mutation operators can be updated as:

$$\begin{cases} pm_1(t) = pm_1 \\ pm_2(t) = pm_2 - \phi(t) \frac{(pm_2 - pm_1)}{2} \end{cases} \quad (10)$$

where pm_1 and pm_2 denote the range of value of the initial mutation probability, and $pm_1(t)$ and $pm_2(t)$ are the corresponding value of the mutation probability of generation t .

When $\phi(t)$ is reduced, the upper limit $\text{pm}_2(t)$ of the mutation probability increases, and the overall variation rate of the population will be promoted. According to the degree of fitness, the individual in the population can be roughly divided into different categories. The roles of each type of individual can behave differently in the process of population evolution, and the corresponding operating probability depends on the specific circumstances of the individual's fitness degree. The adaptability of individuals with better properties is higher, but the general difference between the superior individuals is not large, and the cross-rate can be appropriately reduced.

As a result, the crossover probabilities of two parents for cross-operation in generation t can be given as:

$$\text{pc} = \begin{cases} \text{pc}_2(t) - (\text{pc}_2(t) - \text{pc}_1(t)) \times \frac{f_{\text{avg}}^t - \tilde{f}}{f_{\text{avg}}^t - f_{\text{min}}^t}, & \text{If } \tilde{f} \leq f_{\text{avg}}^t \\ \text{pc}_2(t) - (\text{pc}_2(t) - \text{pc}_1(t)) \times \frac{f_{\text{avg}}^t - \tilde{f}}{f_{\text{max}}^t - f_{\text{avg}}^t}, & \text{Otherwise} \end{cases} \quad (11)$$

where \tilde{f} is the average fitness of the individuals from the generations of the parents.

For individuals who participate in the mutation operation, if the adaptability is higher than the average fitness, a smaller mutation rate should be adopted to prevent the destruction of the outstanding individuals. Otherwise, if the fitness is lower than the average fitness, the mutation rate

should be raised. The detail method is as shown in formula (12).

$$\text{pm} = \begin{cases} \text{pm}_2(t) + (\text{pm}_2(t) - \text{pm}_1(t)) \times \frac{f_{\text{max}}^t - \tilde{f}}{f_{\text{max}}^t - f_{\text{avg}}^t}, & \text{If } f \geq f_{\text{avg}}^t \\ \text{pm}_2(t), & \text{Otherwise} \end{cases} \quad (12)$$

where f denotes the individual's fitness.

5 Simulation results

In order to verify the feasibility and effectiveness of the algorithm, the algorithm proposed in this paper is applied to the optimization problem of typical numerical functions. The experiments are carried out in the VC++6.0 environment, running 100 times, respectively. The population size is 100, and the evolutionary generation is set to 500. In the experiment, three algorithms are used to determine the type of sample selection and single-point crossover, in which the crossover rate of standard genetic algorithm is 0.7 and the variation rate is 0.05. In addition, the crossover rates $\text{pc}_1 = 0.4$ and $\text{pc}_2 = 0.9$, and the mutation probabilities $\text{pm}_1 = 0.01$ and $\text{pm}_2 = 0.1$, respectively. In this section, SGA-IMM is applied to minimize a total of some benchmark functions from the studies [28], as shown in Table 1.

Table 2 shows the experimental results of 100 independent operations on the above functions, including

Table 1 Benchmark functions

Function	Range of the parameters
$f_1(x, y) = 100(x^2 - y)^2 + (1 - x)^2$	$x_1, x_2 \in [-5, 5]$
$f_2(x) = -\sum_{i=1}^{10} x_i \sin(\sqrt{ x_i })$	$x \in [-512, 512]$
$f_3(x_1, x_2) = \cos(5x_1) \cos(5x_2) e^{-0.0001(x_1^2 + x_2^2)}$	$x_1, x_2 \in [-2.048, 2.048]$
$f_4(x_1, x_2) = (4 - 2.1x_1^2 + x_1^4/3)x_2^2 + x_1x_2 + (-4 + 4x_2^2)x_2^2$	$x_1, x_2 \in [-3, 3]$
$f_5(x_1, x_2) = \left\{ \sum_{i=1}^5 i \cos[(i+1)x_1 + i] \right\} * \left\{ \sum_{i=1}^5 i \cos[(i+1)x_2 + i] \right\}$	$x_1, x_2 \in [-10, 10]$
$f_6(x_1, x_2) = 0.5 - \left(\sin^2 \sqrt{x_1^2 + x_2^2} - 0.5 / (1 + 0.001(x_1^2 + x_2^2)) \right)^2$	$x_1, x_2 \in [-10, 10]$
$f_7(x_1, x_2) = 20 + x_1^2 - 10 \cos(2\pi x_1) + x_2^2 - 10 \cos(2\pi x_2)$	$x_1, x_2 \in [-5.12, 5.12]$

Table 2 Experimental results of SGA-IMM on benchmark functions

Function	Global optimal value	Average optimal value	The best value	The worst value	Average convergent generations
f_1	1.0	0.99998	1.0	0.99979	19
f_2	-1.03168	-1.03175	-1.03168	-1.03123	31
f_3	-186.7309	-186.7305	-186.7309	-186.7299	20
f_4	1.0	0.99998	1.0	0.99997	29
f_5	0	6.133E-7	2.377E-8	1.279E-6	459
f_6	-4189.8289	-4189.7322	-4189.8311	-4189.5451	321
f_7	0	5.148E-7	8.455E-7	1.166E-6	525

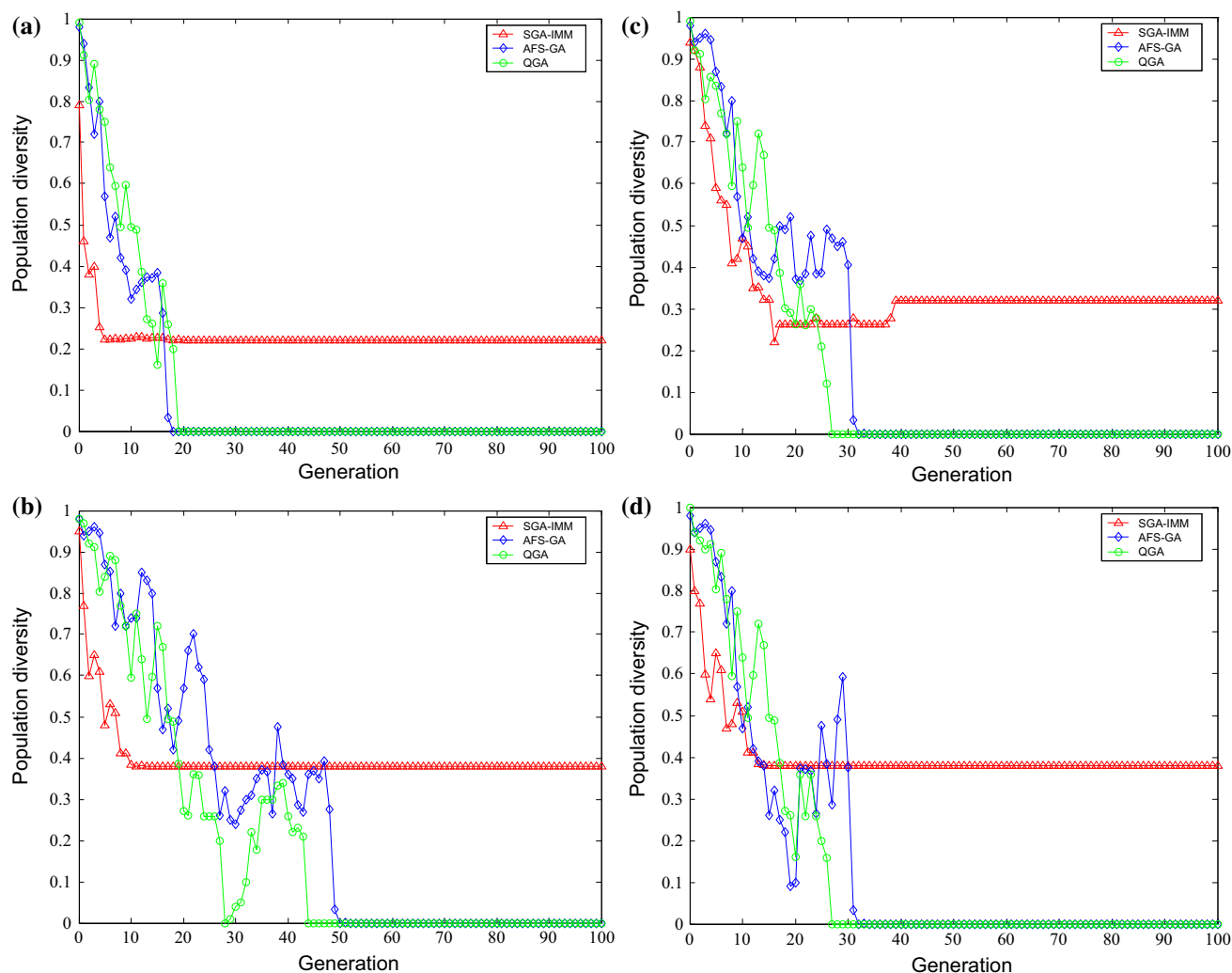


Fig. 1 Comparison of the population diversity. **a** f_1 , **b** f_2 , **c** f_3 , **d** f_4

global optimal value, average optimal value, the best value, the worst value and the average convergent generations from the 100 experiments.

Next, the benchmark functions, such as f_1 , f_2 , f_3 and f_4 , are selected to compare the variation of population diversity, as shown in Fig. 1. As can be seen from Fig. 1, the diversity of the population in AFS-GA [29] and QGA [30] declines to 0 only after less generations of the evolution, which makes the algorithms fall into precocious convergence. SGA-IMM can effectively maintain the diversity of the population, thus effectively avoid the premature convergence and greatly improve the performance of the algorithm.

In order to verify the feasibility and effectiveness of the algorithm, we conduct the experiments in aspects of efficiency and quality. The efficiency of the algorithm is measured by the number of average convergent generations and the number of convergences to the optimal solution. Besides, the number of requisite evaluation times up to

termination criteria is employed to estimate the quality of the algorithm, which includes the mean value, minimum number of times, maximum number of times, and the time consumption, etc. The setting of the experimental parameters and the termination conditions of the algorithm are all consistent with [31]. First, the functions f_1 , f_2 , f_3 and f_4 are selected to evaluate the efficiency of the algorithm. Since the benchmark functions are multimodal, there are many local minimum points near the global optimum value and the algorithms are easy to fall into the local optimum state. The experimental results are shown in Table 3.

Next, the functions f_5 , f_6 and f_7 are applied to test the algorithm's quality. Those functions are representative for performance evaluation of algorithms, which are commonly used by GA. The experimental results are shown in Table 4. From the experimental results, it shows that the average generations of convergence in SGA-IMM are shorter than AFS-GA and QGA in terms of the search efficiency. As for the number of times to search the global

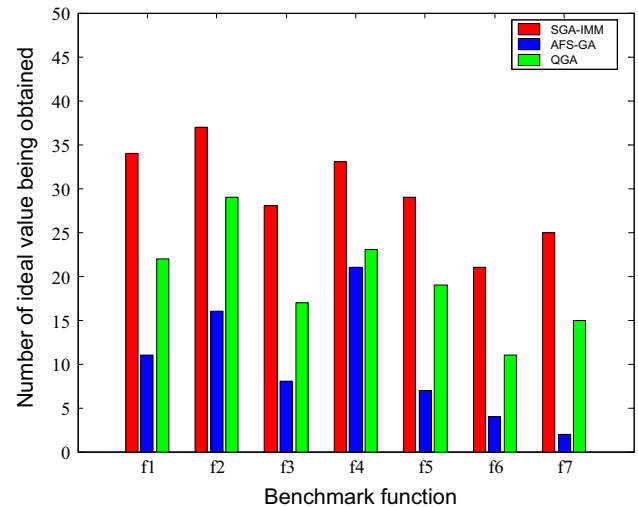
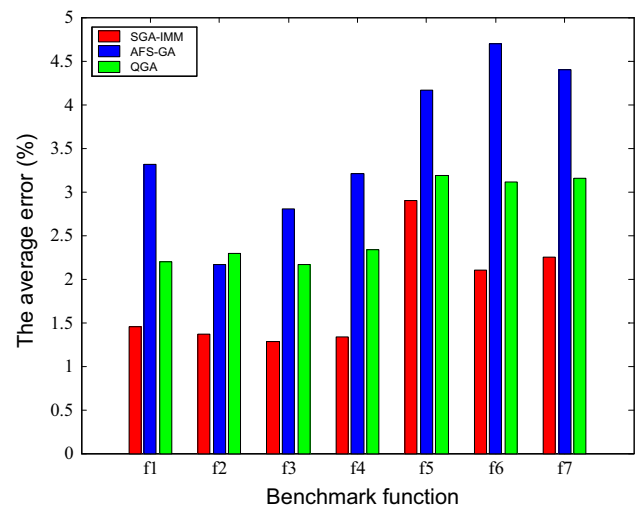
Table 3 Average generations and times for obtaining global optimum value

Function	Algorithm	Average convergent generations	Search times
f_1	SGA-IMM	19	91
	AFS-GA	32	98
	QGA	29	99
f_2	SGA-IMM	31	93
	AFS-GA	58	100
	QGA	35	97
f_3	SGA-IMM	20	92
	AFS-GA	43	99
	QGA	46	100
f_4	SGA-IMM	29	94
	AFS-GA	57	99
	QGA	38	100

optimal value, SGA-IMM also can demonstrate better performance than other algorithms.

Next, the number of maximum generations is set as 1500, and all algorithms run continuously 50 times with regard to different benchmark functions. Figure 2 shows the number of ideal value being obtained during the search procedure. As can be seen from the results, for all benchmark functions, the number of times that the optimal solution equals the ideal value in SGA-IMM is higher than the other two algorithms, especially for functions f_1 , f_2 and f_4 . The reason is that the improved algorithm can maintain the population diversity and improve the genetic algorithm in aspect of the global optimization ability, thus improving the accuracy of solution the problem.

In addition, the average error between the optimal solution and the ideal value at the end of the running time of the algorithm can be obtained. Figure 3 shows the comparison of the average error. Compared to other algorithm, AFS-GA falls into the local optimal lightly, and then the average error of the optimization result deviates much larger than that of the ideal value. SGA-IMM and QGA

**Fig. 2** Comparison of the number of ideal value being obtained**Fig. 3** Comparison of the average error

have great improvement in global search performance than AFS-GA. However, compared with SGA-IMM algorithm, the average error of SGA-IMM algorithm is smaller than that of ideal value, which can achieve higher overall

Table 4 The mean value, minimum number of times and maximum number of times under the termination criteria

Function	Algorithm	Mean value	Minimum number of times	Maximum number of times
f_5	SGA-IMM	571	439	663
	AFS-GA	1184	998	1461
	QGA	820	730	995
f_6	SGA-IMM	677	581	891
	AFS-GA	1378	1153	1429
	QGA	968	874	1154
f_7	SGA-IMM	623	540	722
	AFS-GA	1098	940	1309
	QGA	860	755	1150

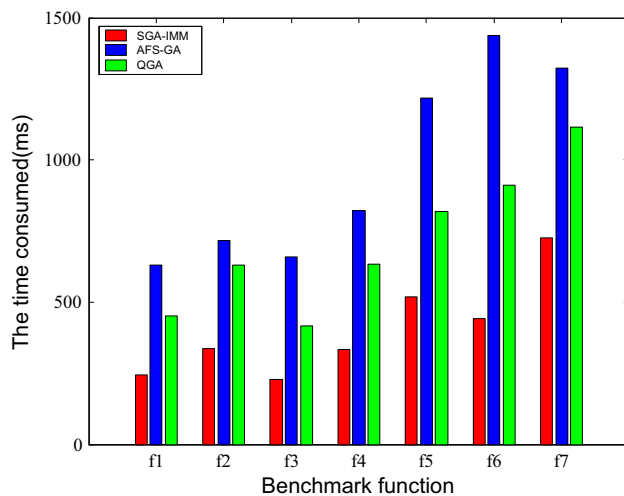


Fig. 4 Comparison of the running time

accuracy of problem solving. Therefore, it demonstrates better performance than QGA in global search performance and evolutionary stability.

Figure 4 shows the comparison of the running time in different algorithms. The SGA-IMM algorithm can better control the crossover and mutation operations according to the current state of the population and avoid premature convergence due to the loss of population diversity. Moreover, the experimental results also show that SGA-IMM can obtain better resolution due to fast optimization within the specified or shorter time intervals. In terms of the quality of the algorithm, the convergence rate of SGA-IMM demonstrates significant superiority compared with AFS-GA and QGA.

6 Conclusions

The premature convergence of GA has unexpected effect on the algorithm's performance, and the main reason is that the evolution of outstanding individuals multiply rapidly will lead to premature loss of population's diversity. To solve the above problem, a method to qualify the population diversity and similarity between adjacent generations is proposed in this paper. According to the evaluation of population diversity and the fitness of individual, the adaptive adjustment of crossover and mutation probability is realized. The experiment results show that SGA-IMM can search the optimal solution of almost all benchmark functions and effectively maintain the diversity of the population. Compared with the existing algorithms, it has greatly improved the convergence speed and the global optimal solution.

Due to the characteristic of randomness in GA, plenty of experiments should be conducted to increase its credibility and reliability. In the future, we will examine to combine other strategies to improve the self-evolution mechanism of GA and enhance the reliability and self-learning of the algorithm.

Acknowledgements This work is supported by the project of the First-Class University and the First-Class Discipline (No. 10301-017004011501). And we wish to thank the anonymous reviewers who helped to improve the quality of the paper.

Compliance with ethical standards

Conflict of interest We declare that we do not have any commercial or associative interest that represents a conflict of interest in connection with the work submitted.

References

1. Choubey NS, Kharat MU (2011) Approaches for handling premature convergence in CFG induction using GA. *Soft Comput Ind Appl* 96:55–66
2. Potter MA, De Jong KA (1994) A cooperative coevolutionary approach to function optimization. In: *Proceedings of the parallel problem solving from Nature-PPSN III, international conference on evolutionary computation*. LNCS 866. Springer, Berlin, pp 249–257
3. Pandey HM, Chaudhary A, Mehrotra D (2014) A comparative review of approaches to prevent premature convergence in GA. *Appl Soft Comput* 24:1047–1077
4. Wang H, Yang S, Ip W, Wang D (2009) Adaptive primal–dual genetic algorithms in dynamic environments. *IEEE Trans Syst Man Cybern Part B Cybern* 39(6):1348–1361
5. Ramadan SZ (2013) Reducing premature convergence problem in genetic algorithm: application on travel salesman problem. *Comput Inf Sci* 6(1):47–57
6. Srinivas M, Patnaik L, Patnaik M (1994) Adaptive probabilities of crossover and mutation in genetic algorithms. *IEEE Trans Syst Man Cybern* 24(4):656–667
7. Mohammadi Ivatloo B, Rabiee A, Soroudi A (2013) Nonconvex dynamic economic power dispatch problems solution using hybrid immune-genetic algorithm. *IEEE Syst J* 7(4):777–785
8. Messelis T, Causmaecker PD (2014) An automatic algorithm selection approach for the multi-mode resource-constrained project scheduling problem. *Eur J Oper Res* 233(3):511–528
9. Hamidreza E, Geiger CD (2008) A fast Pareto genetic algorithm approach for solving expensive multiobjective optimization problems. *J Heuristics* 14(3):203–241
10. Pezzella F, Morganti G, Ciaschetti G (2008) A genetic algorithm for the flexible job-shop scheduling problem. *Comput Oper Res* 35(10):3202–3212
11. Arabali A, Ghofrani M, Etezadi-Amoli M et al (2013) Genetic-algorithm-based optimization approach for energy management. *IEEE Trans Power Deliv* 28(1):162–170
12. Dai XM (2011) Allele gene based adaptive genetic algorithm to the code design. *IEEE Trans Commun* 59(5):1253–1258
13. Zamani R (2013) A competitive magnet-based genetic algorithm for solving the resource-constrained project scheduling problem. *Eur J Oper Res* 229:552–559
14. Lova A, Tormos P, Cervantes M et al (2009) An efficient hybrid genetic algorithm for scheduling projects with resource

- constraints and multiple execution modes. *Int J Prod Econ* 117:302–316
15. Uzor CJ, Gongora M, Coupland S, Passow BN (2014) Real-world dynamic optimization using an adaptive-mutation compact genetic algorithm. In: *Proceedings of 2014 IEEE symposium on computational intelligence in dynamic and uncertain environments (CIDUE)*, pp 17–23
 16. Chen H, Cui DW, Cui YA et al (2010) Ethnic group evolution algorithm. *J Softw* 21(5):978–990
 17. Ling Zhang, Bo Zhang (2001) Good point set based genetic algorithm. *Chin J Comput* 24(9):917–922
 18. Meng W, Han X, Hong B (2006) Bee evolutionary genetic algorithm. *Acta Electron Sin* 34(7):1294–1300
 19. Jiang ZY, Cai ZX, Wang Y (2010) Hybrid self-adaptive orthogonal genetic algorithm for solving global optimization problems. *J Softw* 21(6):1296–1307
 20. Danoy G, Bouvry P, Martins T (2006) Hlcga: a hybrid competitive coevolutionary genetic algorithm. In: *Proceedings of the 6th international conference on hybrid intelligent systems*. Computer Society Press, pp 48–51
 21. Zhou Q, Luo WJ (2010) A novel multi-population genetic algorithm for multiple-choice multidimensional knapsack problems. In: *Proceedings of the 5th international symposium on advances in computation and intelligence*. Springer, Berlin, pp 148–157
 22. Liu Q, Wang X, Fu Q et al (2012) Double elite coevolutionary genetic algorithm. *J Softw* 23(4):765–775
 23. Rojas Cruz JA, Pereira AGC (2013) The elitist nonhomogeneous genetic algorithm: almost sure convergence. *Stat Probab Lett* 83(10):2179–2185
 24. Campos VEM, Pereira AGC (2013) Modeling the genetic algorithm by a non-homogeneous Markov chain: weak and strong ergodicity. *Theory Probab Appl* 57(57):185–192
 25. Hallam J, Akman O, Akman F (2010) Genetic algorithms with shrinking population size. *Comput Stat* 25:691–705
 26. Pereira AGC, Andrade BBD (2015) On the genetic algorithm with adaptive mutation rate and selected statistical applications. *Comput Stat* 30(1):131–150
 27. Jalali Varnamkhasti M, Lee LS, Bakar A (2015) A genetic algorithm with fuzzy crossover operator and probability. *Adv Oper Res* 2012:1687–9147
 28. Sathya SS, Radhika MV (2013) Convergence of nomadic genetic algorithm on benchmark mathematical functions. *Appl Soft Comput* 13(5):2759–2766
 29. Im S-M, Lee J-J (2008) Adaptive crossover, mutation and selection using fuzzy system for genetic algorithms. *Artif Life Robot* 13(1):129–133
 30. Saitoh A, Rahimi R, Nakahara M (2014) A quantum genetic algorithm with quantum crossover and mutation operations. *Quantum Inf Process* 13(3):737–755
 31. Hartmann S (2002) Project scheduling with multiple modes: a genetic algorithm. *Naval Res Logist* 49:433–448