JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 35, 1299-1309 (2019)
DOI: 10.6688/JISE.201911_35(6).0008

Application of Improved Genetic Algorithm in Function Optimization

CHUN YAN[1], MEI-XUAN LI[1] AND WEI LIU[2,+]
[1]*College of Mathematics and System Science*
[2]*College of Computer Science and Engineering*
Shandong University of Science and Technology
Qingdao, 266590 P.R. China
E-mail: liuwei_doctor@yeah.net

In recent years, due to the great potential of genetic algorithms to solve complex optimization problems, it has attracted wide attention. But the traditional genetic algorithm still has some shortcomings. In this paper, a new adaptive genetic algorithm (NAGA) is proposed to overcome the disadvantages of the traditional genetic algorithm (GA). GA algorithm is easy to fall into the local optimal solution and converges slowly in the process of function optimization. NAGA algorithm takes into accounts the diversity of the population fitness, the crossover probability and mutation probability of the nonlinear adaptive genetic algorithm. In order to speed up the optimization efficiency, the introduced selection operator is combined with the optimal and worst preserving strategies in the selection operator. And in order to keep the population size constant during the genetic operation, the strategy of preserving the parents is proposed. Compared with the classical genetic algorithm GA and IAGA, the improved genetic algorithm is easier to get rid of the extremum and find a better solution in solving the multi-peak function problem, and the convergence rate is faster. Therefore, the improved genetic algorithm is beneficial for function optimization and other optimization problems.

Keywords: function optimization, genetic algorithm, global optimization, adaptation, performance simulation

1. INTRODUCTION

The genetic algorithm is an evolutionary algorithm whose basic principle is to choose the population by imitating the theory of species selection and the laws of Darwinian evolution, so as to realize random search and optimization. In recent years, genetic algorithm (GA) has been widely used in various fields [1-3]. But traditional genetic algorithms often trap into local minimum easily when used for the complex functions [4]. Practice shows that it is sometimes difficult to converge to the global optimum for the traditional genetic algorithm. In recent years, in order to improve the performance of genetic algorithm many scholars have proposed various methods of process improvement.

Crossover probability (Pc) and Mutation probability (Pm) were studied as follows. Traditional genetic algorithms use genetic fix operators in the course of genetic evolution [5]. But for the late evolution of the population, genetic fix operators are more likely to destroy better solution. So genetic fix operators will slow the convergence speed of genetic algorithm and lead to premature problem of the algorithm. Feng *et al.* adopted the

Received July 30, 2018; revised November 6, 2018; accepted December 4, 2018.
Communicated by Luís Alberto Lucas.
+ Corresponding author.

adaptive adjustment formula. However, Feng *et al.* neglected that population evolution was unstable and could not be judged by population evolution algebra alone [6]; Tian *et al.* proposed a genetic algorithm that adaptively adjusts mutation probability and crossover probability. However, when the fitness of individuals is lower than the average fitness, the genetic algorithm will adopt larger fixed Pc and Pm values. Larger Pc and Pm values can destroy the high quality genes carried by some undesirable individuals [7]. Yang *et al.* proposed an improved adaptive genetic algorithm (IAGA), which adaptively adjusts the crossover probability and mutation probability according to the degree of population dispersion in evolution [8].

Selection Strategy was studied as follows. Because the selection operation can replicate the individuals with high fitness and eliminate the individuals with low fitness, so can improve the selection strategy to improve the global searching ability of the algorithm [9]. The main idea of the roulette selection operator proposed by Holland 1992, the roulette selection operator is to determine the survival and elimination probability according to the proportion of individual fitness. Because the selection is random, there may be a "degenerate" phenomenon in which good individuals are not selected [10]. Ranjini *et al.* proposed the Elastic Elitism Genetic Algorithm which intelligently conquers the local optimum problem by dynamically controlling the elite count [11]. Kalinli *et al.* proposed a new model based on a dominant gene selection operator for the genetic algorithm [12]. Based on the traditional roulette selection operator Li *et al.* proposed a multi-wheel roulette selection operator based on ranking [13]. Gao *et al.* put forward a sort selection method. This method directly eliminated a quarter of population with low fitness and retained a quarter of population with high fitness [14]. Individual preservation strategies were studied as follows. In this paper, the optimal conservation strategy is to compare the optimal individuals of the old population with the optimal individuals of the new population, and to retain the optimal individuals [8]. This paper puts forward a selection method that combines the sorting selection method with the optimal preservation strategy and worst saving strategy. This method not only preserves the superior individuals, but also ensures the speed of convergence.

Based on the above research, in this paper, a new adaptive genetic algorithm is proposed to solve the problem of invariant crossover probability and mutation probability of traditional genetic algorithm. By analyzing the discrete degree of fitness in the population evolution process, the nonlinear adaptive adjustment genetic algorithm crossover probability and mutation probability value are studied. In order to speed up the evolution of the population and maintain the diversity of the population, this paper uses the sorting selection strategy to eliminate the individuals with low fitness, and directly retains the individuals with high fitness as the parent. In order to keep the population size constant after the selection operation, and to prevent the loss of individuals with higher fitness during the intermediate operation, this paper adds a strategy to preserve the parents in the middle section. In selection operation, the ranking selection strategy is combined with the optimal and worst preserved strategy. The optimal strategy can better preserve the best individuals in history, and the worst individuals can balance the diversity of the population. This not only ensures the preservation of good individuals, but also ensures the convergence efficiency of the algorithm and the equilibrium of population diversity.

In Section 2, we mainly introduce how to improve the genetic algorithm. The section 3, we mainly verify the performance of the improved genetic algorithm in the function optimization. The section 4, we conclude this paper.

2. IMPROVED ADAPTIVE GENETIC ALGORITHM

2.1 Improved Adaptive Genetic Algorithm Flow

The basic genetic algorithm process mainly includes crossover operation, mutation operation and selection operation. Crossover operators are mainly realized by gene exchange among populations, thus achieving the preservation of good genes in populations. If all individuals in the population are missing a good gene, the mutation operator is needed to play a role. The mutation operator achieves the acquisition of the missing good gene by mutation of the individual gene. Crossover operations is first operated in the traditional genetic algorithm, but when the population fitness is small and more concentrated, crossover operation is not conducive to the rapid generation of good individuals. At the later stage of evolution, when the population fitness is large and concentrated, the constant mutation rate will seriously destroy the good individuals of the population and reduce the convergence efficiency. Based on the above analysis, an improved adaptive genetic algorithm called New Adaptive Genetic Algorithm (NAGA) is shown in this paper.

The flow of the improved at this paper are organized as follows.

(1) Code the initial population and set the parameters;
(2) Set the fitness function formula and calculate the fitness value of each individual. And then the maximum fitness individual T1 is retained;
(3) Judge whether the convergence condition is satisfied or not. If the convergence condition is satisfied, output the result, otherwise enter step (4);
(4) If $\pi/12 \leq arcsin(f_{avg}/f_{\max}) < \pi/3$, the mutation operation is performed first, and then the crossover operation is performed, and the parents is retained. Finally, the selection operation is performed. Otherwise, the crossover operation is performed first, then the mutation operation is performed, and finally the selection operation is performed.
(5) If the number of iterations reaches the maximum number of iterations, the iteration stops. Otherwise, Go back to step (2).

The reason why $arcsin(f_{avg}/f_{\max})$ is used as a condition is that with the change of f_{avg}, the change of $arcsin(f_{avg}/f_{max})$ will be faster, so that the degree of centralization and dispersion between population fitness can be better judged. If $arcsin(f_{avg}/f_{\max})$, this indicates is closer to the maximum fitness. Based on Eqs. (3) and (4) the condition $\pi/12 \leq arcsin (f_{avg}/f_{\max}) < \pi/3$, is used to determine whether to cross first or not. If above condition is met the first crossover, or the first mutation operation. In the population, the individual fitness is very small and concentrated, except for the maximum fitness, if and only if satisfies $f_{avg}/f_{\max} < 1/2$. If according to the idea of IAGA algorithm, this situation is classified as population dispersion, crossover operation will be first operated. But in this case, the population difference is not rich, if the first crossover operation will slow down the evolution of the population, this will lead to a slow population convergence. In view of the above situation, this paper improves the condition formula, making the improved algorithm more comprehensive.

2.2 Selection Operator for Improved Adaptive Algorithm

Individuals were sorted according to fitness from large to small. Then, a quarter of population with high fitness were retained and a quarter of population with low fitness were eliminated, and the middle one half of population individuals continued to operate. In this way, both the bad individuals are eliminated and the fine parent is preserved. This not only eliminates the bad parent but also preserves the good parents, and ensures the convergence efficiency of the genetic algorithm, so that the algorithm can quickly find the optimal solution.

Then, the one half of population retained in the previous step calculates the probability of individual selection [13]:

$$\begin{cases} p_k^N = Q^N (1 - q^N)^{k-1} \\ Q^N = \dfrac{q^N}{1 - (1 - q^N)^{L/2}} \end{cases}. \tag{1}$$

p is the selection probability of the individual. k is the ordinal number of an individual in a population, $k = 1, 2, ..., L/2$. L is the number of population. N is number of the current iteration. For the q, in the early stage of population evolution, there are great differences between individuals. Therefore, in order to retain more excellent individuals, individuals with high fitness should also have a large selection probability. With the evolution of population, the differences between populations become smaller and smaller, and the optimal individual selection probability should be reduced. Therefore, a new q value is proposed, which varies according to the number of iterations [13].

$$q^N = q_{\max} - (q_{\max} - q_{\min}) \times \frac{N-1}{M-1} \tag{2}$$

$q_{\max}$ and $q_{\min}$ are the choice probabilities of the best and worst individuals defined at the beginning, M is the total number of iterations. According to the above probability formula, we choose $L/4$ of the new population as part of the parents, and form a new population with $L/2$ number together with the $L/4$ of the first step. In order to keep the population number constant, a method of parent reservation was proposed. In this paper, we use the optimal preservation strategy to preserve the best individuals in the intermediate process, and compare it with the highest fitness T1 of the previous generation. If T1 is higher than the maximum fitness of the new population, an individual of the new population is randomly eliminated, and the highest fitness of the previous generation is added to the new generation to produce a new population. This method ensures that the dominant individuals of the previous generation will not be destroyed by genetic operations such as crossover and mutation. At the same time, in order to balance the diversity of the population, the worst fitness of the population is recorded and preserved. When the worst fitness of the new population is greater than that of the last generation, this paper will randomly eliminate an individual in the new population and add the worst fitness of the previous generation to the new generation to produce a new population. This method not only does not affect the direction of population evolution on a large scale, but also can balance the diversity of the population.

2.3 Adaptive Adjustment of Pc and Pm

In order to give full play to the important role of crossover probability and mutation probability in genetic operation, the adaptive Eqs. (3) and (4) of crossover probability and mutation probability are proposed in this paper [8].

$$P_c = \begin{cases} k_1\left(1 - \dfrac{\arcsin(\frac{f_{avg}}{f_{\max}})}{\pi/2}\right) & \arcsin(\frac{f_{avg}}{f_{\max}}) \ge \pi/6 \\ k_4 \dfrac{\arcsin(\frac{f_{avg}}{f_{\max}})}{\pi/2} & \arcsin(\frac{f_{avg}}{f_{\max}}) < \pi/6 \end{cases} \tag{3}$$

$$P_m = \begin{cases} k_3\left(1 - \dfrac{\arcsin(\frac{f_{avg}}{f_{\max}})}{\pi/2}\right) & \arcsin(\frac{f_{avg}}{f_{\max}}) < \pi/6 \\ k_2 \dfrac{\arcsin(\frac{f_{avg}}{f_{\max}})}{\pi/2} & \arcsin(\frac{f_{avg}}{f_{\max}}) \ge \pi/6 \end{cases} \tag{4}$$

3. VERIFICATION OF IMPROVED GENETIC ALGORITHM IN FUNCTION OPTIMIZATION

3.1 Simulation Experiments and Performance Analysis

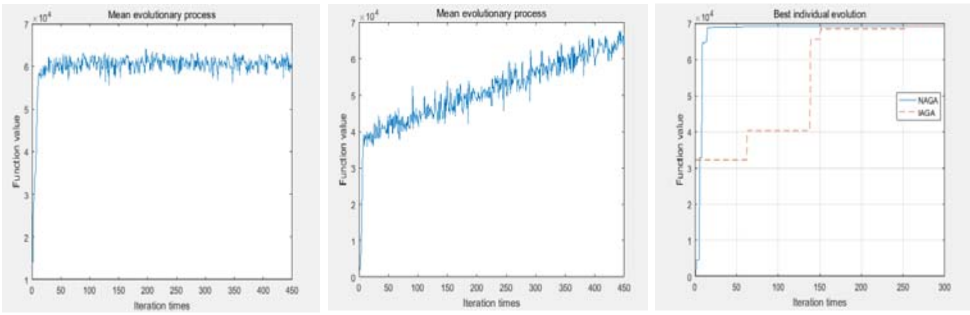
In order to verify the optimization effect of the new adaptive genetic algorithm (NAGA), one-dimensional continuous functions $f_1(x)$ is selected in this paper [8]. Optimal values and convergent times of the two functions are tested. The iterative times and other factors are compared with the classical genetic algorithm GA and Yang's IAGA algorithm. Find the maximum of the function $f_1(x) = 2^{(-2(x-0.1)/0.9)^2}\sin^6(5\pi x) + 7\cos(4x) + 7$, $x \in [0, 2]$.

In order to verify the effectiveness of NAGA algorithm in solving multidimensional functions, NAGA algorithm is compared with particle swarm optimization (PAO) and ant colony algorithm (ACO) in multi-dimensional function $f_2(x)$, $f_3(x)$ and $f_4(x)$. $f_2(x) = \sum_{i=1}^{n} |x_i| + \prod_{i=1}^{n} |x_i|$, $-2 \le x_i \le 2$, n = 2, 5, 10. $f_3(x) = 1 + \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos(\frac{x_i}{\sqrt{i}})$, $-2 \le x_i \le 2$, n = 2, 5, 10. $f_4(x) = \sum_{i=1}^{n}[x_i^2 - 10\cos(2\pi x_i) + 10]$, $-2 \le x_i \le 2$, n = 2, 5, 10.

3.2 Solving Function $f_1(x)$ with Three Algorithms

GA and IAGA and NAGA all use binary coding. The parameters were set follows. In formula $f_1(x)$, chromosome length is 32 bits. The number of iterations is 450 generations and population size is 100. The crossover probability and mutation probability of GA algorithm are 0.6 and 0.01 respectively.

By calculating the maximum value of the function $f_1(x)$ in the case of $x \in [0, 2]$, the new genetic algorithm proposed in this paper is compared with the GA and IAGA algorithm. As can be seen from Figs. 1 (a) and (b), the average population fitness of GA is close to the maximum population fitness, which makes the new solution unable to get rid of the local extremum. While the average fitness of the population generated by the NAGA algorithm is gradually improved, and the average values are not all concentrated near the larger values. This not only guarantees the diversity of the population, but also ensures that the population is not easy to fall into the local maximum. The NAGA algorithm adaptively increases the crossover probability and reduces the mutation probability when the population average is not close to the maximum fitness. Through adequate chromosome exchange, the destruction of good individuals can be reduced. The NAGA algorithm has strong search ability.



(a) The result of mean value of GA.
(b) The result of mean value of NAGA.
(c) Comparison of simulation results of IAGA and NAGA based on function $f_1(x)$.

Fig. 1. Simulation result diagram based on function $f_1(x)$.

From Fig. 1 (c), one can find the IAGA algorithm only uses the optimal retention strategy to keep the optimal solution, its convergence speed is obviously less than the NAGA algorithm. The NAGA algorithm combines the best and the worst preservation strategy. NAGA algorithm not only uses the optimal selection algorithm to retain 1/4 individuals with large fitness. Moreover, we use the worst preservation strategy to save one of the worst individuals. This not only improves the speed of population evolution, but also balances the population diversity. In order to directly observe the ability of each genetic algorithm, three algorithms are repeated 30 times. Then the average iteration times, optimal values, convergence probability and convergence times of the three algorithms are compared. The results, as showed in Table 1.

Table 1. The result of calculation of function $f_1(x)$.

Algorithm	The optimal solution	Convergence degree	Convergence times	Convergence probability
GA	6.899940357765080e+04	417	14	0.447
IAGA	6.899941721276939e+04	251	25	0.833
NAGA	6.899941721750828e+04	50	30	1

Table 1 shows the optimal solution obtained by NAGA algorithm is better than GA and IAGA algorithm, so the improved algorithm has higher accuracy. The number of iterations decreased from 417 times (GA) and 251 times (IAGA) to 50 times (NAGA). Therefore, the improved algorithm proposed in this paper converges faster. In addition, after 30 repeated tests, the average convergence times of the three algorithms are increased from 14 (GA) and 25 (IAGA) to 30 (NAGA). Which shows that the improved genetic algorithm has better stability. The results in Table 1 show that the solution obtained by NAGA after convergence is superior to that obtained by GA and IAGA. Therefore, GA and IAGA do not find the optimal solution when convergence occurs. Therefore, the genetic algorithm and IAGA algorithm fall into local extremum. They can't get rid of the local extremum, so we can't find the best value.

3.3 Solving Multidimensional Functions with Five Algorithms

In order to verify the effectiveness of NAGA algorithm in solving multidimensional functions, NAGA algorithm is compared with particle swarm optimization (PSO) and ant colony algorithm (ACO) in multi-dimensional function $f_2(x)$, $f_3(x)$ and $f_4(x)$. The population size of each algorithm is 100, and the maximum number of iterations is 600 times. In genetic algorithm, mutation probability is 0.05, crossover probability is 0.7. The pheromone Volatilization Coefficient and transfer probability constant of ant colony algorithm are 0.8 and 0.2 respectively, and the learning factor of particle swarm optimization is 1.49445. Each algorithm runs 30 times independently for each test function, and then obtains the optimal value, the worst value, the standard deviation and the mean convergence algebra (MCA) of each function. The specific results are shown in Tables 2, 3 and 4. Figs. 2-4 are simulated pictures of five optimization functions at $f_2(x)$, $f_3(x)$ and $f_4(x)$ at $n = 5$.

From the comparison of Tables 2, 3 and 4, we can see that the optimal value, worst value, standard deviation and average convergence algebra obtained by NAGA algorithm are smaller than those obtained by GA, IAGA, ACO and PSO algorithms. Taking function $f_2(x)$ as an example, the optimal value of function solved by NAGA algorithm is 9.313225748323190e-10 in 2-dimensional case, which is much smaller than ACO and

Table 2. The result of calculation of function $f_2(x)$.

Dimension	Algorithm	Optimal value	Worst value	standard deviation	MCA
2	GA	9.313225748323190e-10	6.053596739151587e-06	1.376711701389571e-06	419.2
	IAGA	9.313225748323190e-10	2.328306448703445e-08	4.974508104619992e-09	524.23
	PSO	1.266404336525856e-05	8.944264329630682e-04	2.264486177013865e-04	430.73
	ACO	8.429240665235797e-07	6.693584802801848e-05	1.505079966253631e-05	472.07
	NAGA	9.313225748323190e-10	9.313225748323190e-10	0	51.80
5	GA	4.041003995582937e-04	0.007535471117016	0.001795045933530	512.4
	IAGA	1.193429343682162e-04	0.001575146336475	3.771218334428711e-04	567.4
	PSO	0.001721985838132	0.010893586440201	0.002620249266148	484.33
	ACO	8.351667765734251e-04	0.498748652273138	0.122299734029746	579.93
	NAGA	4.190951585769653e-09	5.685724318027496e-07	1.048191850564316e-07	491.93
10	GA	0.039440487520639	0.095536060653518	0.014830395778162	562.03
	IAGA	0.020122783263243	0.086047551614709	0.016063883807190	558.03
	PSO	0.020496745657030	0.054089103635652	0.007864213335203	556.2
	ACO	0.664931371530980	3.086295002314504	0.480979429994666	598.07
	NAGA	7.997453214625416e-05	6.778426469953303e-04	1.316253238873309e-04	543.5

Table 3. The result of calculation of function $f_3(x)$.

Dimension	Algorithm	Optimal value	Worst value	standard deviation	MCA
2	GA	0	2.919309238791357e-11	7.115128643267135e-12	288.5
	IAGA	0	5.551115123125783e-16	1.503720245745664e-16	367.4
	PSO	1.963629259194022e-10	4.681982923582240e-08	1.492394187377681e-08	327.37
	ACO	2.433386825373418e-12	1.206367450379275e-09	2.230330235398854e-10	495.4
	NAGA	0	0	0	47.83
5	GA	1.780183785937695e-07	2.080190187359055e-05	5.580319979859862e-06	539.17
	IAGA	2.131165355301334e-09	1.948364080073262e-07	4.289209240153261e-08	578.277
	PSO	2.451403314784884e-07	4.013875895436669e-06	9.251730684628982e-07	538.47
	ACO	3.801251435842090e-08	0.001963507858619	5.646640707723747e-04	506.63
	NAGA	1.110223024625157e-16	8.781864124784988e-14	1.774238216373729e-14	525.03
10	GA	7.555026338468274e-05	0.001547641424453	3.818532188984084e-04	549.73
	IAGA	4.466270948888518e-05	2.717404152944303e-04	7.282570467968450e-05	576.37
	PSO	8.190479271474871e-06	6.140052730241319e-05	1.324724669414902e-05	554.17
	ACO	0.002534973173269	0.140250841818148	0.034073721205463	599.1
	NAGA	1.667113558312394e-09	2.508355978481092e-08	5.543823044867356e-09	553.93

Table 4. The result of calculation of function $f_4(x)$.

Dimension	Algorithm	Optimal value	Worst value	standard deviation	MCA
2	GA	0	3.730349362740526e-14	7.477500263014978e-15	472.3
	IAGA	0	7.874010776731666e-10	1.630490304723725e-09	296.3
	PSO	6.794961393552512e-08	2.744419133193787e-05	6.985301971985145e-06	380.9
	ACO	1.397772919631279e-08	2.625598275685093e-06	5.993536477112387e-07	437.63
	NAGA	0	0	0	36.7
5	GA	3.603857514633546e-06	1.985039690310941e-04	3.928060494510266e-05	574.97
	IAGA	5.890932186503051e-10	0.001802156943818	4.528279673150857e-04	451.47
	PSO	0.001340781052429	4.975598174292236	1.001078192849794	524.33
	ACO	3.937777226070693e-05	1.990103890366777	0.710754732837850	516.07
	NAGA	0	4.229505634611996e-12	1.050745858271823e-12	434.33
10	GA	0.045233831526675	1.193134111366309	0.257442531931293	566.17
	IAGA	1.982983469375199e-04	2.752524382023628	0.617490921714679	573.57
	PSO	1.013929491189716	11.011656331583424	2.328717502747265	557.6
	ACO	1.990933310312919	6.965249794556537	1.072764673306460	570.4
	NAGA	7.346382879802604e-08	1.562540853861094e-05	3.427501291042656e-06	559.73

PSO algorithm. Although the results are the same as those of GA and IAGA, it can be seen from Fig. 2 that the convergence algebra of GA and IAGA is obviously higher than that of NAGA. Moreover, the standard deviation of GA and IAGA is obviously larger than that of NAGA. It shows that NAGA is more stable than the other four algorithms. For $f_3(x)$ and $f_4(x)$ functions. The optimal solution, the worst solution and the standard deviation of the function obtained by NAGA in 2-D are all 0. And the convergence speed of NAGA is the smallest. This shows that the stability, accuracy and convergence speed of NAGA algorithm are obviously better than those of the other four algorithms. From the simulation comparisons of the test functions in Figs. 2, 3 and 4, we can see that NAGA algorithm converges downward rapidly and converges faster so it shows that compared with the other four algorithms, NAGA algorithm has stronger global searchability. And the optimal solution after NAGA convergence is lower than the other four algorithms, which shows that the NAGA algorithm has good local search ability and high efficiency.

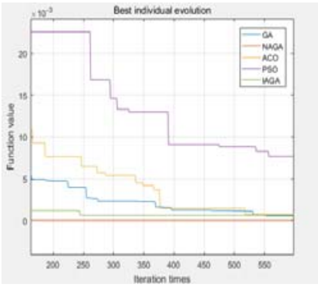


Fig. 2. Simulation result diagram based on function $f_2(x)$.

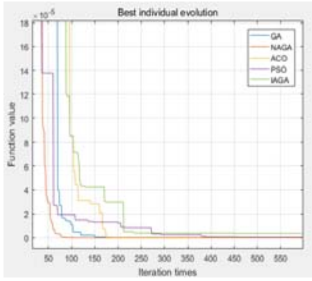


Fig. 3. Simulation result diagram based on function $f_3(x)$.

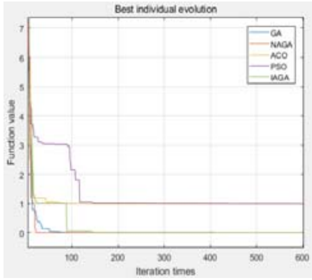


Fig. 4. Simulation result diagram based on function $f_4(x)$.

4. CONCLUSION

With the rapid development of computer technology, genetic algorithm will be more widely used in various fields of national economy. With the in-depth study of genetic algorithm and its fusion with other disciplines, it will bring vitality and motivation in the field of artificial intelligence and promote the progress of human beings. The NAGA algorithm presented in this paper can not only effectively grasp the global evolution direction, but also overcome the shortcomings of poor global search ability and immature convergence of genetic algorithm. In order to overcome the shortcomings of GA algorithm such as poor global search ability and immature convergence, NAGA algorithm adaptively adjusts the crossover probability and mutation probability. This paper improves the flow conditions of IAGA algorithm and takes into accounts many possible situations of fitness distribution. On this basis, a selection operator and an optimal preservation strategy are introduced to record and preserve individuals with high fitness. This ensures that good genes in the intermediate process are not destroyed and that individuals with low fitness are eliminated in time. Finally, the simulation results show that compared with the GA algorithm, the convergence speed and accuracy of NAGA algorithm in solving univariate and multivariate functions have been greatly improved. However there are limitations. The NAGA algorithm requires the population fitness to be positive. Although it does not affect the calculation effect, the objective function needs to be adjusted properly in order to solve the optimization problem. Therefore, the algorithm can be further improved so that fitness can be negative.

ACKNOWLEDGMENT

This work was financially supported by the Project of National Natural Science Foundation of China (No. 61502280, 61472228), the Project of Qingdao Applied Basic Research of Qingdao (special youth project, No. 14-2-4-55-jch).

REFERENCES

1. M. Moussa, M. Hmila, and A. Douik, "A novel face recognition approach Based on genetic algorithm optimization," *Studies in Informatics & Control*, Vol. 27, 2018,

pp. 127-134.
2. A. Moeini and M. R. Zargarabadi, "Genetic algorithm optimization of film cooling effectiveness over a rotating blade," *International Journal of Thermal Sciences*, Vol. 125, 2018, pp. 248-255.
3. A. M. Abdelsalam and M. A. El-Shorbagy, "Optimization of wind turbines siting in a wind farm using genetic algorithm based local search," *Renewable Energy*, Vol. 123, 2018, pp. 748-755.
4. S. Li, C. Zhao, X. Su, *et al.*, "Temperature and humidity control for environmental test chamber based on genetic algorithm optimized parameters of PID controller," *Journal of Nanjing University of Science and Technology*, Vol. 41, 2017, pp. 511-518.
5. Y. Q. Zhao and Q. Qian, "Adaptive genetic algorithm based on optimal weighted evolution and its performance research," *Information Technology*, Vol. 11, 2018, pp. 68-72.
6. L. H. Feng and F. Yang, "Fuzzy PID control of pneumatic shield gate based on improved genetic algorithm," *Water Conservancy Planning and Design*, Vol. 12, 2017, pp. 98-101.
7. X. Tian, G. R. Liu, W. B. Zhou, and K. F. Guo, "Research on robot path planning based on improved adaptive genetic algorithm," *Machine Tools and Hydraulics*, Vol. 44, 2016, pp. 24-28.
8. C. R. Yang, Q. Qian, F. Wang, and M. H. Sun, "Application of improved adaptive genetic algorithm in function optimization," *Computer Application Research*, Vol. 35, 2018, pp. 1042-1045.
9. W. Khan, A. Salhi, M. A. Jan, and R. A. Khanum, "New version of a multi-algorithm genetically adaptive for multiobjective optimization," *Applied Soft Computing*, Vol. 39, 2016, pp. 292-309.
10. J. H. Holland, "Adaptation in natural and artificial systems," *Ann Arbor*, Vol. 6, 1992, pp. 126-137.
11. A. Ranjini and B. S. E. Zoraida, "Impact on various selection operators in elastic elitism genetic algorithm for solving energy consumption scheduling in smart home," *Journal of Advanced Research in Dynamical and Control Systems*, Vol. 9, 2017, pp. 602-618.
12. A. Kalinli and Ö. Aksu, "Genetic algorithm model based on dominant gene selection operator," *Journal of the Faculty of Engineering and Architecture of Gazi University*, Vol. 26, 2011, pp. 869-875.
13. C. Li and H. Y. Ning, "Improved genetic algorithm selection operator," *Journal of Tianjin University of Technology*, Vol. 24, 2008, pp. 1-4.
14. H. Gao and N. Y. Xue, "Fitting LED spectrum model with back propagation neural network based on improved genetic algorithm," *Progress in Laser and Optoelectronics*, Vol. 54, 2017, pp. 294-302.

Chun Yan (闫春) received the B.S., M.S. and Ph.D. degrees from Shandong University of Science and Technology, Qingdao, China, in 2000, 2003 and 2011, respectively. She is presently an Associate Professor of Shandong University of Science and Technology, Qingdao, China. Her research interests include applied mathematics, statistics and economic management. She has about 20 technical papers published in journals and conference proceedings in his research areas.

Mei-Xuan Li (厉美璇) received the B.S. degree from Ludong University, China, in 2017. She is now pursuing the M.S. degree in the Shandong University of Science and Technology, Qingdao, China. Her research interests include applied mathematics and optimization theory.

Wei Liu (刘伟) received the B.S., M.S. and Ph.D. degrees from Shandong University of Science and Technology, Qingdao, China, in 1999, 2005 and 2012, respectively. He is presently a Professor of Shandong University of Science and Technology, Qingdao, China. His research interests are in formal engineering methods, Petri nets, workflows, and web services. He has led or participated in many projects supported by the National Natural Science Foundation, National Key Basic Research Development Program, and other important and key projects at provincial levels. He has published more than 30 papers in journals and conference proceedings, such as *IEEE Transactions on Systems Man and Cybernetics: Systems* and *Enterprise Information Systems.* He was the recipient of Science and Technology Advancement Awards at the province level and the Excellent Doctoral Dissertation Award from Shandong province, China.