An improved adaptive genetic algorithm considering the overall process solving capacity

LIU Zhi-jie [a] *, Zhao Xiao-qian[b], LIU Xiao-long[c], LIU Xiao-yu[d]

Transportation Equipment and Ocean Engineering College, Dalian Maritime University, Dalian 116026, China

[a]liuzj2003@163.com, [b]zhaoxq616@126.com, [c]lxl241@163.com, [d]liuxiaoyu9326@hotmail.com

Keywords: Adaptive Genetic Algorithm (AGA); overall evolutionary capacity; solving stability

Abstract. To solve the problems of easily falling into local optimal solution and solving process instability for adaptive genetic algorithms, an improved adaptive genetic algorithm is presented by improving the crossover and mutation probability calculation method, which balances the solving stability and reliability in the early and late process. The crossover and mutation probability of the superiority individuals should be relatively small in the evolutionary process. But they can not be zero so as to avoid early entering into local optimal solutions since the superiority individuals can not make the crossover and mutation operation and produce new individuals in early evolutionary process. The worse individuals should have a greater crossover and mutation probability in the evolutionary process in order to increase the number of new individuals and prevent premature entry into the local optimal solution. The case analysis demonstrates that this algorithm has better ability of global optimization and stability.

Introduction

Genetic algorithm (GA) is simple and easy to use, has good robustness and is applicable to solve complex mathematical modeling problems, and is widely used in solving engineering problems [1, 2]. But basic GA is easy to enter the premature and only obtains the local optimal solutions since the crossover and mutation probability are constant in the evolutionary process and it ignores the individual adaptive change ability with the change of the environment [3]. The adaptive genetic algorithm can change the crossover and mutation probability, population number or the gene encoding length with the individual fitness value change. This can solve the premature phenomenon of the basic GA in a certain extent. For example, Srinivas [4] and Chen [5] et al. present representative adaptive genetic algorithm. But the experimental results show that the traditional adaptive genetic algorithm still exist some problems of easily falling into local optimal solution or solving process instability. This paper analyzes the influence rules of the crossover and mutation probability on the solving process and proposes an improved adaptive genetic algorithm to improve solving quality and stability by balancing the algorithm solving performance in the whole process.

Design principle of the algorithm

Analysis of some typical adaptive genetic algorithms. The crossover and mutation probability affect the GA's behavior and performance in the computational process. The bigger the crossover probability P_c is, the faster GA generates new individuals. But if a big crossover probability is chosen in the later stage of evolution, the higher fitness parent individuals will be broken and reduce their probability. The bigger the mutation probability P_m is, the faster GA generates new individuals, too. In the early stage of evolution, with few higher fitness individuals, the mutation probability should increase to some degree in order to expand the search scopes into the entire region quickly. However in the later stage of revolution, for higher-fitness individuals, the mutation probability should be decreased to some degree so as to avoid the lost of genes in good parent individual chromosomes. So appropriate crossover and mutation probability should be chosen in order to improve the stability of in overall evolution solving process.

Srinivas et al. [4] first put forward to the adaptive genetic algorithm S-AGA. In this algorithm, when the fitness values gradually tend to the maximum fitness value, the P_c and P_m gradually become small and eventually become zero. The above P_c and P_m meet the solving requirements in the later stage of revolution. But in the early evolution stage, the above P_c and P_m will make the excellent individuals in initial population almost fall into a constant state. So it is easy to fall into the local optimal solution. Chen et al. [5] present an adaptive genetic algorithm P-AGA based on superiority inheritance. This definition will affect the solving stability for crossover probability P_c when it is close to the optimal solution in the late stage of evolution.

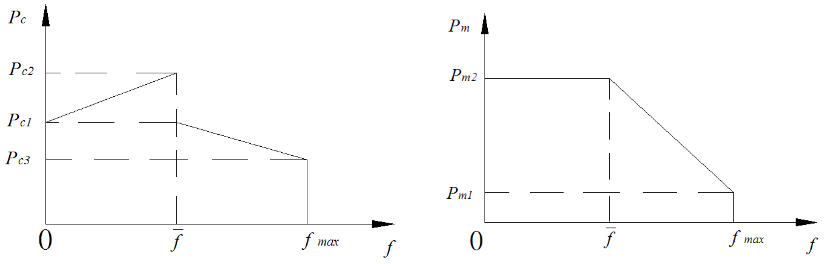
An improved adaptive genetic algorithm. The good or bad individuals should both have a bigger crossover and mutation probability in order to prevent individuals from falling into the local optimal solutions in the initial stage of evolution. In the later stage of evolution, the crossover and mutation probability of good individuals is small and the bad individual is big. So the crossover and mutation probability of improved adaptive genetic algorithm (G-AGA) are as follows:

$$
P_{\mathrm{c}} = \begin{cases} P_{\mathrm{c1}} + \dfrac{(P_{\mathrm{c2}} - P_{\mathrm{c1}})}{f_{\max} - \bar{f}} \cdot f' & f' \le \bar{f} \\ P_{\mathrm{c3}} + \dfrac{(P_{\mathrm{c1}} - P_{\mathrm{c3}})}{f_{\max} - \bar{f}} \cdot (f_{\max} - f') & f' > \bar{f} \end{cases}
\qquad (1)
$$

$$
P_{\mathrm{m}} = \begin{cases} P_{\mathrm{m2}} & f'' \le \bar{f} \\ P_{\mathrm{m1}} + \dfrac{P_{\mathrm{m2}} - P_{\mathrm{m1}}}{f_{\max} - \bar{f}} \cdot (f_{\max} - f'') & f'' > \bar{f} \end{cases}
$$

Where f' denotes the bigger fitness value of two crossed individuals; f'' denotes the fitness value of mutation individuals; $f_{\max}$ denotes the maximum fitness value of the population; $\bar{f}$ denotes the average fitness value. The individuals whose fitness values are higher than the average fitness value are called good individuals. Else, they are called bad individuals.

The change curves of the crossover and mutation probability for the G-AGA are shown in Fig.1. When the individual is good one, the crossover and mutation probability should be relatively small in the evolutionary process, but not to zero. If it is zero, the superiority individual will not make the crossover and mutation operation and does not produce new individuals in early evolutionary process, which will fall into local optimal solution untimely. The bad individuals should have a greater crossover and mutation probability in the overall evolutionary process in order to increase the number of new individuals and to prevent its entry into the local optimal solution untimely.



a) Crossover probability b) Mutation probability

Fig. 1 Adaptive crossover and mutation probability curve for G-AGA

The computing process of the proposed adaptive genetic algorithm G-AGA is shown in Fig. 2.

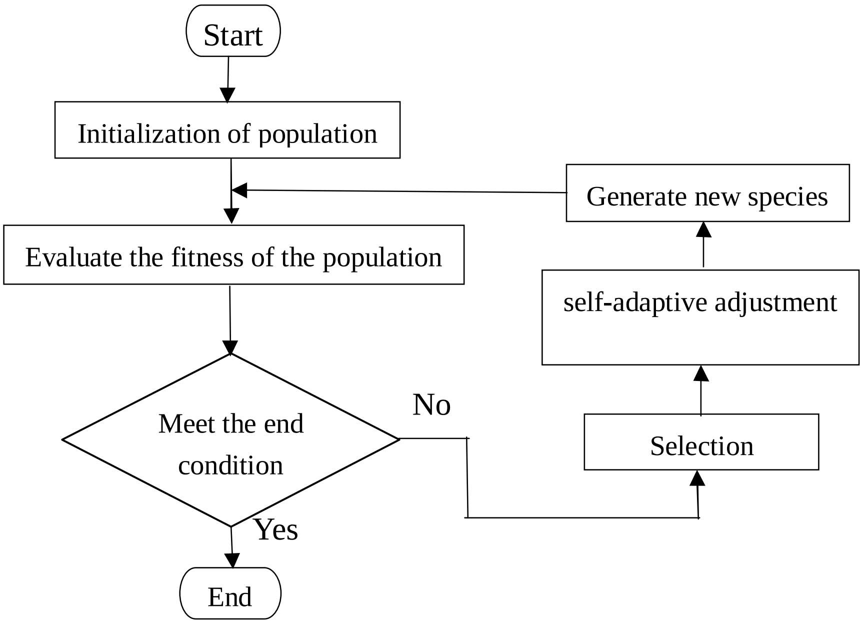


Fig. 2 Programming flowchart of G-AGA

Example calculation and result analysis

This paper adopts the following test functions to evaluate the solving performance of G-AGA.

$$\begin{cases} f(x) = x^2 - 10\cos(2p\,x) + 10 \\ 0 \le x \le 4 \end{cases} \tag{2}$$

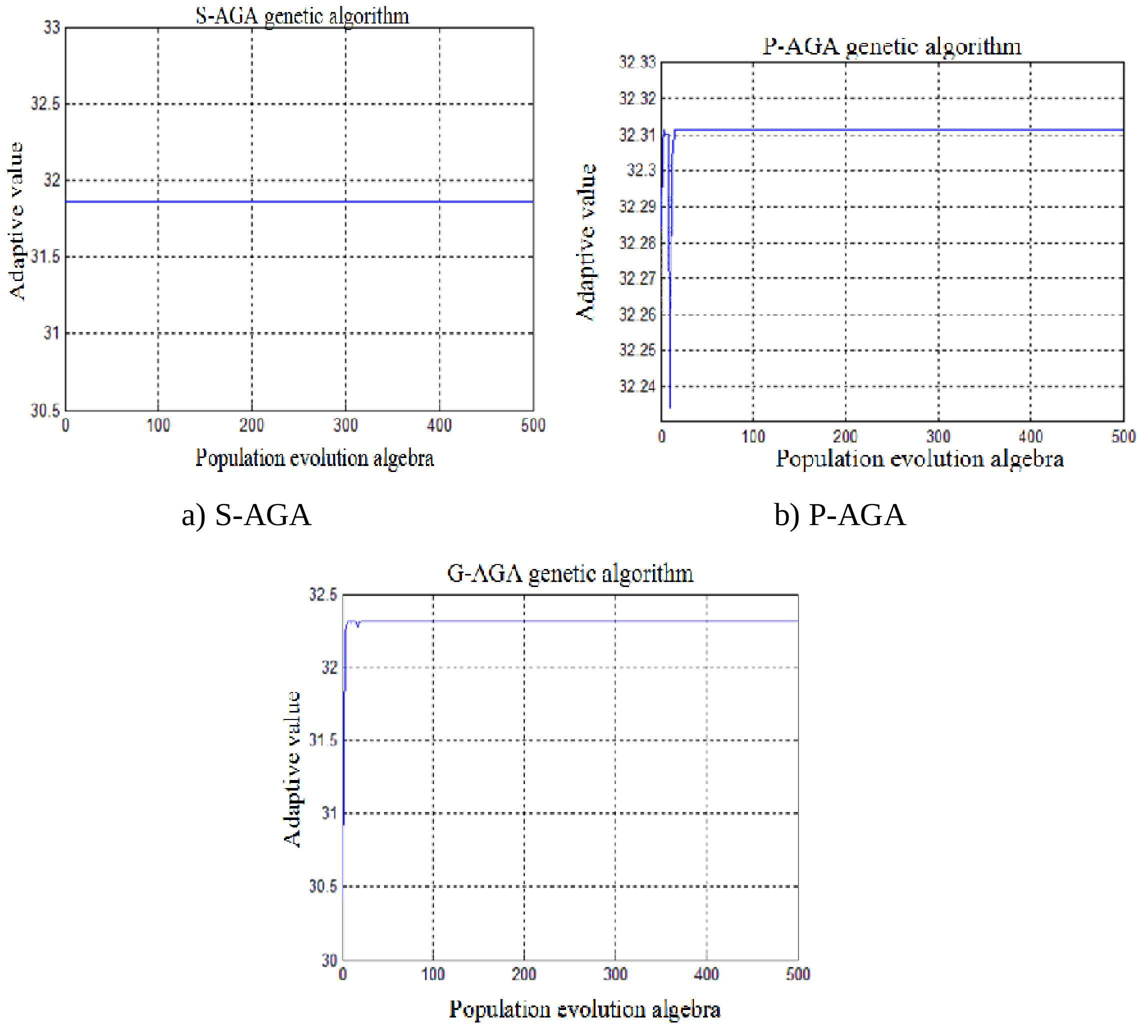
$$\begin{cases} f(x) = 0.5 + \dfrac{\sin^2\sqrt{x^2 + \dfrac{p^2}{36}} - 0.75}{\left[0.998 + 0.001\left(x^2 + 2\right)\right]^2} \\ -100 \le x \le 100 \end{cases} \tag{3}$$

The initial population is 50 and the maximum evolution algebra is 500. The test makes five experiments for each algorithm. P_{c1} =0.6, P_{c2} =0.9, P_{c3} =0.2, P_{m1} =0.1 and P_{m2} =0.001. The optimization results are as shown in Tab. 1. From Tab. 1, the G-AGA can obtain better optimization results and good solving quality.

Tab. 1 The test experimental results for the case

Formula	S-AGA		P-AGA		G-AGA	
	x	*f*	*x*	*f*	*x*	*f*
2	3.4989	32.2421	3.4989	32.2421	3.5116	32.3048
	3.5018	32.2622	3.4794	32.0220	3.5194	32.3120
	3.4911	32.1720	3.5028	32.2681	3.5145	32.3103
	3.5292	32.2875	3.5106	32.3022	3.5214	32.3100
	3.5282	32.2916	3.5096	32.2993	3.5184	32.3124
3	1.5141	0.7479	-1.3706	0.7384	89.4554	0.5030
	7.8030	0.7211	14.1835	0.6711	-92.2104	0.5005
	-7.7282	0.7123	-1.5652	0.7425	-97.8325	0.4949
	-7.6245	0.6839	-4.5437	0.7217	-81.8259	0.4877
	7.9190	0.7154	-1.5217	0.7474	-83.9759	0.4971

In the optimization process, the changes of the fitness values with the evolution algebra in the case are as shown in Fig. 3 and Fig. 4. From Fig.3 and Fig.4, we can know the solving process early falls into the local optimal solution and can not obtain the global optimal solution. The solving process fluctuates due to the function of crossover and mutation probability. The proposed G-AGA has better stability in the process of solving two test functions optimization.

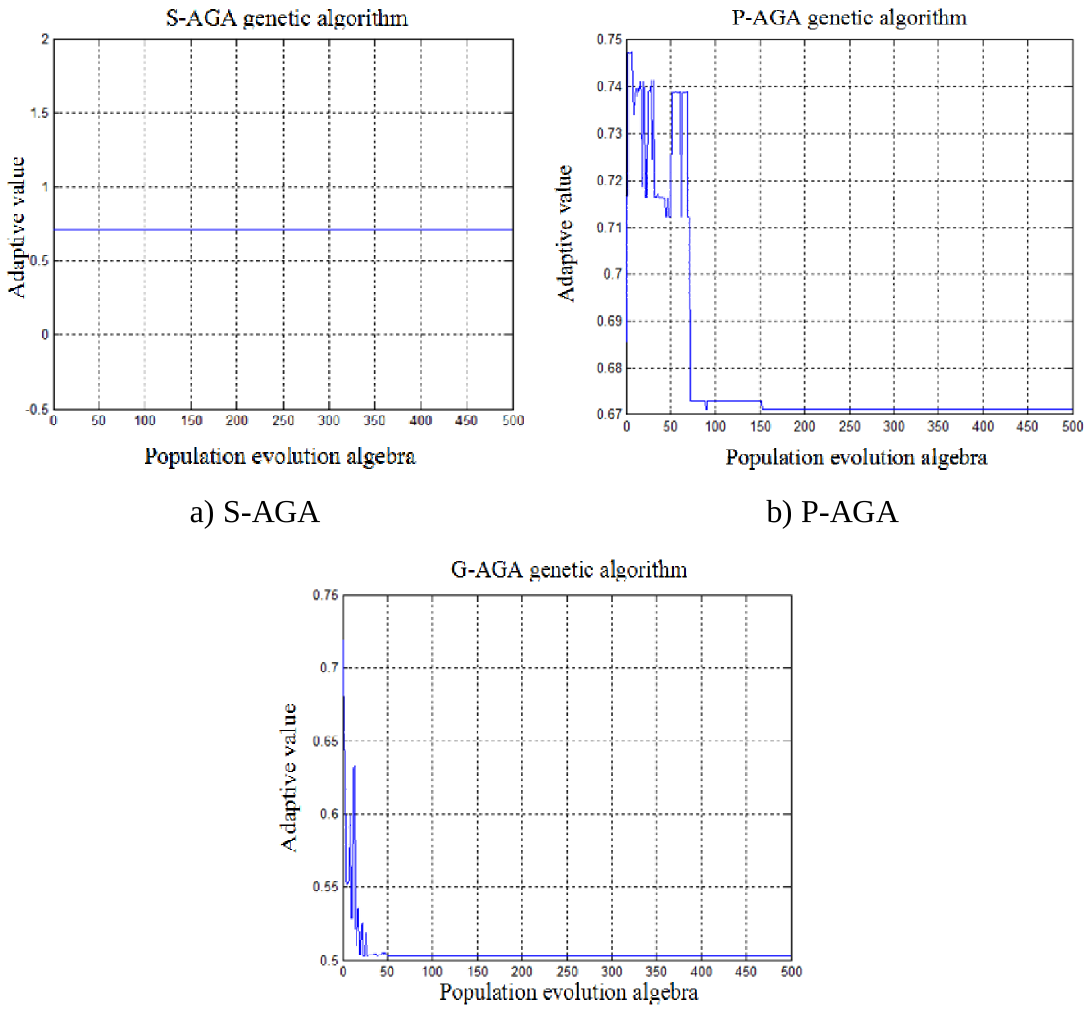


a) S-AGA

b) P-AGA

c) G-AGA

Fig. 3 Fitness values changes with the evolution algebra for solving Eq. (4)



a) S-AGA

b) P-AGA

c) G-AGA

Fig. 4 Fitness values changes with the evolution algebra for solving Eq. (5)

Conclusions

Based on the analysis of the previous adaptive genetic algorithms, and considering the evolutionary performance in the whole process, an improved adaptive genetic algorithm is presented. This algorithm improves the crossover and mutation probability so as to improve the solving quality and stability. The example analysis shows that the algorithm is of the high solving accuracy, good stability and strong capability of global search. This will benefit improving the solving reliability of the AGA.

Acknowledgments

The authors are grateful to the financial support by Research Fund for the Doctoral Program of Higher Education of China (20122125120013), Scientific Research Fund of Liaoning Provincial Education Department (L2013206), Natural Science Foundation of Liaoning Province(2015020121), the Fundamental Research Funds for the Central Universities (3132015087).

References

[1] Leao F. B., Pereira R. A. F., Mantovani J. R. S. Fast fault section estimation in distribution

control centers using adaptive genetic algorithm[J]. Electrical Power and Energy Systems, 2014, 63:787-805.

[2] Chen R., Liang C. Y., Hong W. C. et al. Forecasting holiday daily tourist flow based on seasonal support vector regression with adaptive genetic algorithm[J]. Applied Soft Computing, 2015, 26:435-443.

[3] Lim T. Y., Al-Betar M. A., Khader A. T. Adaptive pair bond in genetic algorithm: An application to real-parameter optimization[J]. Applied Mathematics and Computation, 2015, 252:503-519.

[4] Srinivas M, Patnaik L M. Adaptive probabilities of crossover and mutation in genetic algorithms[J]. IEEE Tranns on Systems, Man and Cybemetics, 1994, 24(4):656-667.

[5] Chen Ghizhe, Liu Guodong, Pu Xin et al. Adaptive genetic algorithm based on superiority inheritance[J]. Journal of Harbin Institute of Technology, 2007,39(7):1021-1024.