World Academy of Science, Engineering and Technology
International Journal of Mathematical and Computational Sciences
Vol:9, No:12, 2015

Application of Adaptive Genetic Algorithm in Function Optimization

Panpan Xu, Shulin Sui

***Abstract*—**The crossover probability and mutation probability are the two important factors in genetic algorithm. The adaptive genetic algorithm can improve the convergence performance of genetic algorithm, in which the crossover probability and mutation probability are adaptively designed with the changes of fitness value. We apply adaptive genetic algorithm into a function optimization problem. The numerical experiment represents that adaptive genetic algorithm improves the convergence speed and avoids local convergence.

***Keywords*—**Genetic algorithm, Adaptive genetic algorithm, Function optimization.

I. Introduction

IN the field of artificial intelligence, we need to find the optimal solution in complex search space to solve function optimization and combinatorial optimization problems [1]. In recent years, people pay more attention to the function optimization problems with the continuous development of computer application technology. However, it is difficult to find a common method to solve complex function optimization problems because of the expansion of scale and increased constraints.

As an intelligent algorithm, genetic algorithm has been widely used in function optimization field. Genetic algorithm (GA) starts the search from a group instead of a single initial solution to get the optimal solution and it has a good capability of parallel processing [2]. However, with the depth of applied research, genetic algorithm exposes shortcomings, such as the low search speed, low search precision and premature convergence. More and more people are working on the study of improving genetic algorithm. The study found that improving crossover probability and mutation probability can overcome these disadvantages.

II. Function Optimization

Optimization theories and methods have been a hot topic because of its wide applicability and practicability in recent years. It is an optimal solution research on the mathematical definition. Combined with the actual problem, it selects appropriate methods and strategies from a number of programs to find the optimal solution [3]. Many practical problems in real life can be abstracted as function optimization problems.

The typical function optimization model can be described as $Max\{f(X) \mid X \in S\}$. $X = (x_1, x_2, ..., x_n)^T$ represents a set of decision variables, $x_i (i = 1, 2, ..., n)$ are generally in the real domain R, $f(X)$ is the objective function of this model, S is a subset of n dimensional Euclidean space R^n, S is generally described by the equation or inequality with a group of decision variables. For example:

$$\begin{aligned} &Min \quad f(X) \\ &s.t. \quad c_i(X) \geq 0 (i = 1, 2, ...k_1) \\ &\qquad c_i(X) = 0 (i = k_1 + 1, k_1 + 2, ..., k) \end{aligned}$$

$c_i(X) \geq 0 (i = 1, 2, ..., k_1)$ and $c_i(X) = 0 (i = k_1 + 1, k_1 + 2, ..., k)$ are constraints. Feasible solution is the point in R^n that meets the constraints.

$$S = \{X \in R^n \mid c_i(X) \geq 0 (i = 1, 2, ..., k_1), c_i(X) = 0 (i = k_1 + 1, k_1 + 2, ...k)\}$$

is the feasible region that is a collection of all the feasible solutions.

For $\forall X \in S$, $f(X^*) \geq f(X)$ then $X^* \in S$ is a global optimal solution in this optimization model $Max\{f(X) \mid X \in S\}$, and $f(X^*)$ is the global optimal value of this optimization model $Max\{f(X) \mid X \in S\}$. If there is $\delta > 0$,

$$\forall X \in S \cap \{X \in R^n \mid \sqrt{\sum_{i=1}^{n} (x_i - x_i^*)^2} < \delta\}, f(X^*) \geq f(X)$$

then $X^* \in S$ is the local optimal solution of this optimization model [4]. Global optimal solution must be local optimal solution, but local optimal solution is not necessarily the global optimal solution.

Due to the expansion of the scale and increased constraints, it is difficult to find a common method to solve complex function optimization problems. So we need to search a method and find the approximate optimal solution at acceptable time and accuracy. Genetic algorithm has many advantages, such as universal and parallelism [5], [6]. Genetic algorithm is a more popular bionic optimization algorithm and it has been widely applied to the function optimization problems.

III. Introduce of Adaptive Genetic Algorithm

A. Principles of Adaptive Genetic Algorithm

Genetic Algorithm (GA) is a kind of simulated evolutionary algorithm which can effectively solve the function optimization problems. Numerous studies find that environment of

P. Xu and S. Sui are with the Department of Mathematics, Qingdao University of Science & Technology, Shandong 266061, China (e-mail: qustxupanpan@163.com).

parameters is essential to the quality of genetic algorithm performance and results. The changes of parameters can affect the capabilities of function optimization. However, genetic algorithm often uses fixed crossover probability and mutation probability in the practical applications, the crossover probability and mutation probability remain unchanged throughout the search process. So it is difficult that genetic algorithm converges to the global optimal solution in solving complex optimization problems [7]. Adaptive genetic algorithm (AGA) which is submitted by M. Srinivas mainly focuses on adaptively designing the formulas of crossover and mutation probabilities. And the crossover probability and mutation probability are adaptively designed with the changes of fitness values to preserve the best individual of the population, and it can prevent premature.

The definitions of p_c and p_m are as [8]:

$$p_c = \begin{cases} k_1 \dfrac{f_{max} - f'}{f_{max} - f_{ave}}, & f' \geq f_{ave} \\ k_3 & f' < f_{ave} \end{cases}$$

$$p_m = \begin{cases} k_2 \dfrac{f_{max} - f}{f_{max} - f_{ave}}, & f \geq f_{ave} \\ k_4 & f < f_{ave} \end{cases}$$

where f_{avg} is the average fitness value of all individuals in population, f_{max} is the largest individual fitness value in population, f' is the larger fitness value of the two crossed individuals, f is fitness value of individual which will do mutation operation, k_1, k_2, k_3, k_4 is between 0 and 1, k_3 is larger than k_1 and k_4 is larger than k_2, after setting the values of k_1, k_2, k_3, k_4, the crossover probability and mutation probability may be adaptively adjusted. Assume $k_1 = k_2 = k, k_3 = k_4 = k'$, The adaptive drawing line of crossover probability is shown in Fig. 1 and adaptive drawing line of mutation probability is shown in Fig. 2.

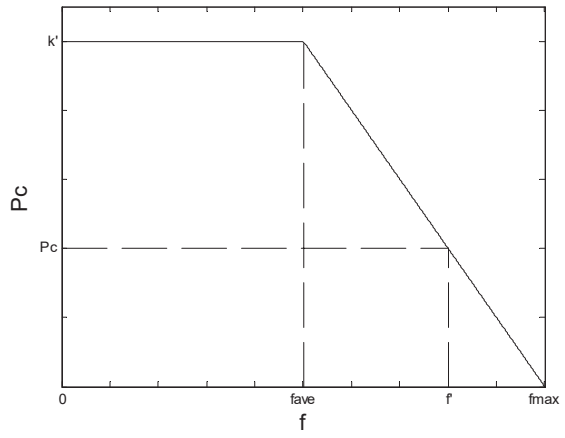

Fig. 1 Adaptive drawing line of crossover probability in adaptive genetic algorithm

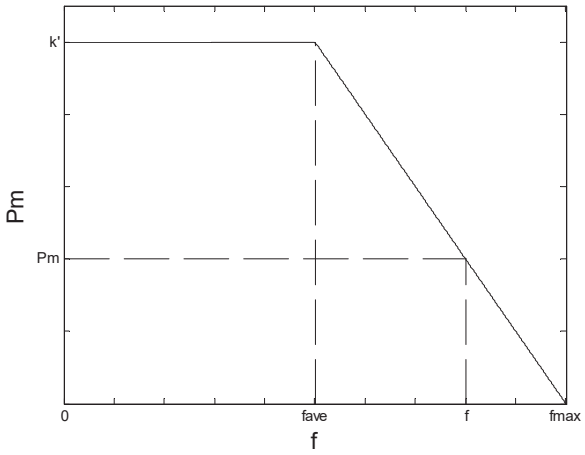

Fig. 2 Adaptive drawing line of mutation probability in adaptive genetic algorithm

Figs. 1 and 2 represent that the crossover probability and mutation probability do linear transformation with the changes of individual fitness values. We use a larger p_c and p_m for individuals of which fitness values are less than the average fitness. For individuals of which fitness values are larger than the average fitness, p_c and p_m adaptively changes with the change of fitness function [9].

B. Steps of Adaptive Genetic Algorithm

The basic steps of adaptive genetic algorithm are as follows:

(1) Set population size N, hereditary frequency T, then code gene and randomly generate initial population P.
(2) According to the fitness function, calculate the fitness of individuals in population.
(3) Select individual. The individual with higher fitness has more chance to be selected and copied.
(4) Calculate crossover probability according to the formula that is given in adaptive genetic algorithm. Do crossover operation on individuals in population in accordance with certain principles.
(5) Calculate mutation probability according to the formula that is given in adaptive genetic algorithm.
(6) Do mutation operation based on the adjusted formula and the mutation principle, generate a new population.
(7) Determine whether the algorithm meets the termination iterations, if it does not meet the termination iterations, then do step (2).
(8) Output results.

Adaptive genetic algorithm is an improvement on the traditional genetic algorithm [10]. Genetic algorithm generates a new generation of individuals by using crossover operation which is essentially the process of genetic recombination. If the crossover probability is too large, the individuals of population update faster, and the individuals with high fitness values will soon be destroyed, which means the fine mode will soon be destroyed. If the crossover probability is too small, genetic recombinant will be reduced, ultimately leading to searching stagnation, and genetic algorithm does not converge.

Mutation operation is essentially the disturbance of

population mode in genetic algorithm and mutation operation can increase the diversity of the population. But if the mutation probability is too small, it will be difficult to produce a new model, if the mutation probability is too large, then the search will be more random.

The crossover probability and mutation probability do not remain fixed in adaptive genetic algorithm, and it greatly improves the convergence accuracy and rate of genetic algorithm by adjusting adaptively genetic parameters.

C. Optimal Preservation Strategy

A real problem of using adaptive genetic algorithm is that the crossover probability and mutation probability may become large at some point, which can damage the good individuals [11]. In this paper, we adopt the optimal preservation strategy to preserve the best individual. If the highest fitness value of a new generation is smaller than the highest fitness value of the last generation, we randomly eliminate an individual of the new generation, and add an individual with the highest fitness value of the last generation. Optimal preservation strategy ensures that current best individual cannot be damaged by the crossover and mutation operations.

IV. Numerical Experiment

To verify the performance of AGA, we select a representative function to experiment compared with GA.

Seek maximum of the typical function:

$$f(x_1,x_2)=100(x_1^2-x_2)^2+(1-x_1)^2, x1, x2\in[-2.048,2.048]$$

this function has two local maxima at point （2.048，-2.048） and （-2.048，-2.048）. Its geometry is shown in Fig. 3.

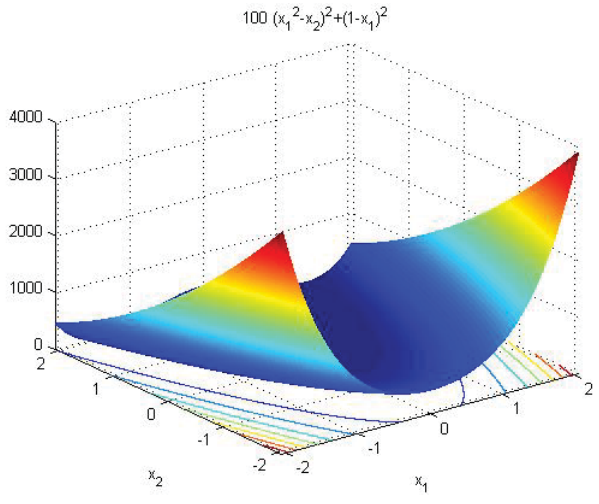

Fig. 3 Function geometry

We experiment the above function for 50 times separately with different p_c and p_m (other parameters are consistent). The results are shown in Table I:

TABLE I
The Performance of GA with Different Crossover and Mutation Probabilities

test function	$f(x_1,x_2)=100(x_1^2-x_2)^2+(1-x_1)^2$			
p_c	0.6	0.75	0.8	0.85
p_m	0.02	0.05	0.03	0.05
convergence times	24	19	26	25
convergence probability	48%	38%	52%	50%

We can see from Table I that different crossover and mutation probabilities may affect the results of the optimization. Table I represents that experiment does not converge to the global optimum every time. GA leads into a local optimum easily if the crossover and mutation probability remain unchanged throughout the search process.

The above function is tested separately by GA and AGA, in which the population size and the maximum evolution generation are the same, The parameters are supposed with $k_1=0.5, k_2=0.02, k_3=0.85, k_4=0.05$, $p_m=0.05, p_c=0.85$. We program by software MATLAB, and the results are shown in Table II. Table II represents that we can get the maximum value of a function by AGA, which is better than GA.

TABLE II
The Results Obtained by 2 Algorithms

algorithm	maximum	x_1	x_2
GA	3834.7	-2.0449	-2.0036
AGA	3905.9	-2.048	-2.048

The simulation diagram of objective function with GA is shown in Fig. 4 and the simulation diagram of objective function with AGA is shown in Fig. 5.

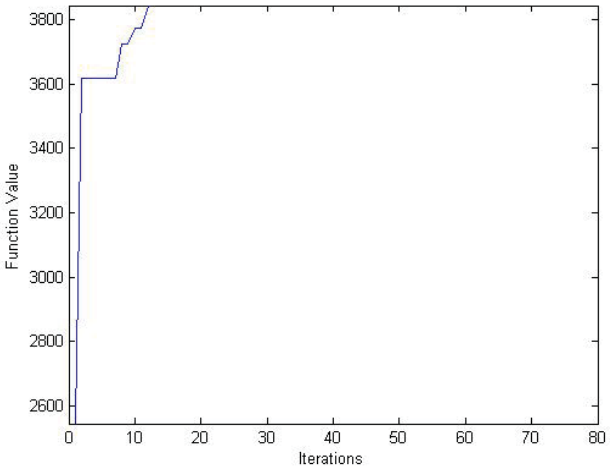

Fig. 4 Optimization results of GA

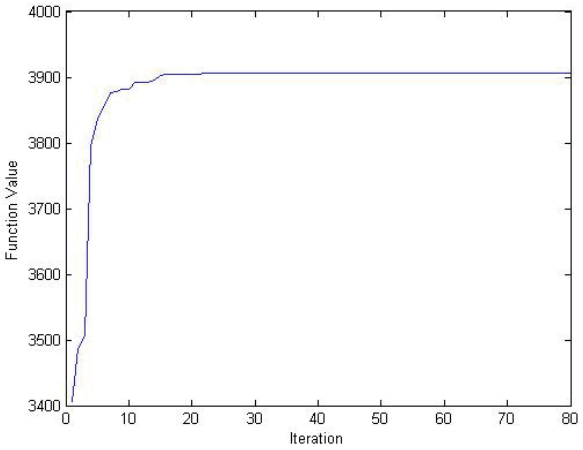

Fig. 5 Optimization results of AGA

Fig. 4 represents that GA has a long time stagnation in the search process, and this phenomenon often happens in the early evolution. When the individuals of population appear more modes, it is difficult to let these individuals change the old patterns as soon as possible in order to escape from local convergence because of the fixed crossover probability and mutation probability [12]. After many generations, the algorithm does not find the global optimum. We can see from Fig. 5, AGA shows a better performance that adapts surroundings in the optimization process, the convergence rate of AGA is significantly better than GA, and AGA can find the global optimum.

V. Conclusion

Crossover probability and mutation probability are no longer fixed in adaptive genetic algorithm and the two important parameters can change with the values of fitness in the evolutionary process. We validate superiority of adaptive genetic algorithm by numerical experiment and it proves that adaptive genetic algorithm effectively solve the typical function optimization problems.

References

[1] Y.X. Yuan, W.Y. Sun. Optimization Theory and Methods, Beijing: Science Press, 2007, pp.1-50.
[2] L.Y. Jia, X. Du. Study of Parallel Genetic Algorithm, Journal of Hunan City University, 2006, 15(3), pp.72-74.
[3] X.L. Wang, J. lu. Optimization Methods and Optimal Control, Harbin: Harbin Engineering University Press, 2006, pp. 3-74.
[4] J. Liu, W.C. Zhong, F. Liu. Organizational Evolutionary Optimization, Journal of computers, 2004, 27(2), pp. 157-167.
[5] Y. Zeng. Application of Improved Genetic Algorithm in Nonlinear Equations, Journal of East China Jiaotong University, 2004, 21(4), pp.39-41.
[6] Y.F. Sun, Z.J. Wang. Application of Genetic Algorithm in Function Optimization Progress, Control and Decision, 1996, 11(4), pp. 425-431.
[7] G.Y. Liao. Adaptive Genetic Algorithm, Technology Square, 2007(3), pp. 70-72.
[8] C.Z. Chen, N. Wang. Adaptive Approach and Mechanism of Crossover and Mutation Probability in genetic algorithm, Control Theory & Applications, 2002, 19(1), pp. 41-43.
[9] J.Z. Zhang, T. Jiang. Improved Adaptive Genetic Algorithm, Computational Engineering and Applications, 2010, 46(11), pp.53-55.
[10] Z.W. Ren, Z. San. Improved Adaptive Genetic Algorithm and Its Application in System Identification, Journal of System Simulation, 2006, 18(1), pp. 41-43.
[11] L. Zhang, Y. Liu, W. He. Application of Adaptive Genetic Algorithm in License Plate Location, Computer Application, 2008, 28(1), PP. 185-188.
[12] W.L. Wang, Q. D. Wu, Y. Song. Adaptive genetic algorithm shop scheduling problem, Systems Engineering Theory and Practice, 2004, 12(2), pp. 58-62.