IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 18, Issue 1, Ver. IV (Jan – Feb. 2016), PP 73-77
www.iosrjournals.org

An Improved Genetic Algorithm Based on Adaptive Differential Evolution

Feng Jiang[1,3], Yi Shen[1,2,3,*], Panpan Ge[2], Cheng Zhang[1,3], Mingxin Yuan[1,2,3]
[1]*(Suzhou Institute of Technology, Jiangsu University of Science and Technology, Zhangjiagang, China)*
[2]*(School of Mechanical and Power Engineering, Jiangsu University of Science and Technology, Zhangjiagang, China)*
[3]*(Service Center of Zhangjiagang Camphor Tree Makerspace, Zhangjiagang, China)*

Abstract: *In order to solve the premature convergence and improve the optimization ability of simple genetic algorithm (SGA) in the complex function optimization, an improved differential evolution based-genetic algorithm (DGA) is proposed in this paper. On the basis of the selection and crossover operation of SGA, the adaptive differential evolution operator is used as the main optimization method to realize the mutation operation. Furthermore, the optimal reservation and worst elimination strategies are added in the SGA. All of these improvements can significantly improve the optimization performance of SGA. Compared with the SGA, the simulation results of function optimization show that the proposed DGA is characterized by strong global optimization ability, quick convergence speed and good stability, which shows the validity of the improvements to simple genetic algorithm.*

Keywords: *Self-adaption, Differential evolution, Genetic Algorithm, Function optimization*

I. Introduction

Genetic algorithm is an adaptive global optimization probability search algorithm, which simulates the biological genetic and evolution process in the natural environment [1]. A general framework for solving the optimization problem of complex systems is provided, which does not depend on the specific areas of the problems and has strong robustness to various types of problems. An improved genetic algorithm based on arithmetic crossover operator and non-uniform mutation operator is proposed in [2], which can avoid the premature convergence, but the search ability is slightly less. The crossover operator and the simulated annealing algorithm are used to allow the parent generation to participate in the competition in [3], which can improve the global search ability and help it quickly escape from the local minima, but its optimization performance is sensitive to the initial temperature of the simulated annealing algorithm. Differential evolution is a kind of heuristic random search algorithm based on group difference and the optimization is finished using real encoding in continuous space. But it is incapable of "survival of the fittest" during the selection process. The mutation operation in the algorithm only aims at the newly generated individual. Besides, all the individuals in the new generation have no equal chance of mutation, so it is easy to trap into the local optimal solutions. Compared with the genetic algorithm, the mutation operator of differential evolution algorithm is more important, and its mutation probability is much higher than that of genetic algorithm. It is easy to see that the differential evolution algorithm and the genetic algorithm have strong complementarity. Aiming at the problems, such as premature convergence, weak search ability, in complex function optimization of genetic algorithm, inspired by the basic principles of differential evolution and the evolution mechanism of genetic algorithm, an improved genetic algorithm based on adaptive differential evolution is designed in this paper, and the advantages of genetic algorithm and differential evolution algorithm are fully used to improve the optimization ability of intelligent algorithm. The function optimization results also verify the validity of the proposed DGA.

II. Simple Genetic Algorithm

The simple genetic algorithm mainly includes encoding, initial population generation, fitness evaluation, selection, crossover, mutation and other operations. The process can be described as follows:

(1) Code all possible solutions to the problem;
(2) find an objective fitness function for the problem optimization;
(3) Generate an initial population that satisfies all constraints;
(4) calculate the fitness of each chromosome in the population;
(5) Assess whether the termination condition is met. If not, proceed with the following process, otherwise output the optimal solution and end;
(6) execute the selection operation, that is to generate a new population according to the fitness of each chromosome;

[*] **Corresponding author.** E-mail: sheny456@hotmail.com

(7) Execute the crossover and mutation operation. The probabilities of crossover and mutation operation are denoted as $\boldsymbol{P_c}$ and $\boldsymbol{P_m}$ respectively;

(8) Assess whether the termination condition is met again. If not, turn to step (6), otherwise output the optimal solution and end.

The workflow of the simple genetic algorithm is shown in Fig.1.

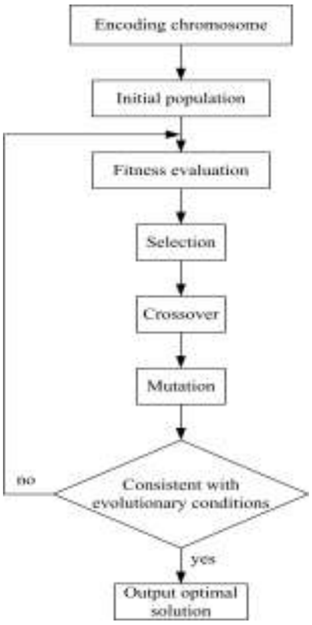

Fig. 1 Workflow of simple genetic algorithm

III. Design of Genetic Algorithm Based on Adaptive Differential Evolution

As mentioned above, the DGA is designed on the basis of the selection and hybridization operation of SGA and the adaptive differential evolution operator is used to realize the mutation operation. Furthermore, the optimal reservation and worst elimination strategies are added in the SGA.

Population initialization and coding

In the DGA, according to the range of variables and optimization precision, 20 individuals are randomly generated to form the initial population. In order to facilitate the subsequent operations, the binary encoding is used in DGA.

Fitness evaluation and detection

Not only fitness is used to measure the quality of each individual in the population, but also a measure of each individual's ability of adaptation to its environment. In the algorithm, the chromosome encoding of each individual can be achieved after binary encoding. An individual is a possible solution to the practical problem, and all the possible solutions correspond with the function values. Here, the function refers to the fitness function, and the function value is the fitness value. In intelligent algorithms, the fitness is a very important index, whose size directly affects the survival probability of each individual in the population. It also has a great influence on the convergence speed and other optimization performance of the algorithm. Therefore, the fitness function should be carefully designed. In this paper, because the fitness function is designed according to the objective function, it also reflects the problem to be solved.

Selection strategy

Before selecting operation, according to the principle of the best reservation and the worst elimination, the individuals in the population are pretreated. By setting the threshold value of the reservation ratio and the elimination ratio, the individuals whose fitness is greater than $\boldsymbol{f_a}$ are directly retained, and the individuals whose individual fitness is less than $\boldsymbol{f_b}$ are directly eliminated. Then the proportional selection operation is used to select the operation. The individual selected probability can be described as:

$$P_i = \frac{f_i}{\sum_{1}^{N} f_i} (i = 1,2,3,......, N) \tag{1}$$

where f_i is the fitness value of individual *i*. *N* is the size of the population.

The probability that an individual is selected is proportional to its fitness. And the selection operation can improve the average fitness of the population at the same time. New population is composed of direct retention of individuals and selected individuals.

Crossover operator

The execution of crossover operator is: First, two individuals (namely X_i, X_j) are randomly selected to make a pair, then whether the crossover operation of the two selected individuals is carried out depends on the pre-set hybrid probability $\boldsymbol{P_h}$. That is to say, if a uniform random number which is generated in (0, 1) is less than $\boldsymbol{P_h}$, the crossover operation is carried out, and two new individuals (namely $X_i^{'}$, $X_j^{'}$) are generated. The crossover process can be described as follows:

$$\begin{cases} X_i' = X_i(1,2,\cdots,\mu) + X_j(\mu+1,\cdots,L) \\ X_j' = X_i(\mu+1,\cdots,L) + X_j(1,2,\cdots,\mu) \end{cases} \tag{2}$$

Where μ is the crossover point, L is the total length of binary encoding.

Adaptive differential evolution operator

In difference strategy[5], three points from the current population are arbitrarily selected. One point is taken as a base point and the other two points are taken as reference points, then a disturbance is generated. The generation process of the disturbance can be described as:

$$V_n(m+1) = y_{r1}(m) + \delta(y_{r2}(m) - y_{r3}(m)) \quad n \neq r_1 \neq r_2 \neq r_3 \tag{3}$$

where δ is a scaling factor and $y_n(m)$ represents the *n*th individual in the *m*th generation.

The scaling factor δ is dynamically adjusted along with the individual fitness and evolution generation. When the individual fitness in the population tends to converge or converge to the local optimal solution, the δ is increased. When the individual fitness in the population is relatively dispersed, the δ decreases. At the same time, if the fitness is greater than the average fitness of the population, corresponding to the larger δ, the individual is eliminated. In contrast, the individuals who are below average fitness and closer to the average fitness of the population correspond to the smaller δ to ensure population diversity. Therefore, the adaptive scaling factor is able to provide the best δ to each solution. The introduced adaptive difference operation maintains the population diversity in the algorithm. At the same time, the convergence of the algorithm is also guaranteed. The δ is on the basis of the following Eq.(4) to carry out adaptive adjustment:

$$\delta = \begin{cases} 1 - \dfrac{f_{avg} - f'}{f_{avg} - f_{best}} & f' < f_{avg} \\ 1 & f' \geq f_{avg} \end{cases} \tag{4}$$

Where f' is the fitness value of an individual. f_{avg} is the average fitness value of current population. f_{best} is the maximum fitness value of the current population.

At the beginning of the optimization, because the gap between f_{best} and f_{best} is very large, there is no possibility of local convergence. If the f' is smaller, the δ is smaller, then good genes can be preserved. With the increase of the evolution generation, the difference of the gap between f_{best} and f_{best} becames smaller and δ tends to decrease. That is to say that the speed of convergence of the optimal solution is gradually accelerated. Because the convergence speed is gradually accelerated, the risk of local convergence is reduced, and the global search ability is improved.

Algorithm flow

In order to further improve the optimization performance of the algorithm, based on SGA, the crossover operator and the adaptive differential evolution operator are introduced. The algorithm flow of the DGA is as follows:

Step1 Initialize algorithm parameters: population size $\boldsymbol{N}$, reservation ratio threshold f_a , elimination ratio threshold f_b, maximal evolution generation k_{max}, individual selected probability P_i, crossover probability P_h, variable precision V_p, and variable range V_r.

Step2 Generated the initial population on the basis of variable precision and variable range.

Step3 Execute adaptive evaluation and detection and generate the population according to the principle of the best reservation and the worst elimination. Calculate the individual selection probability by Eq.(1), and execute the selection operation.

Step4 Execute crossover operation according to the probability P_h using Eq.(2).

Step5 Execute adaptive differential evolution operation using Eq.(3) and new population is got. The scaling factor δ is dynamically adjusted according to Eq.(4).

Step6 Judge whether the terminating condition is satisfied. If not, k←k+1, go to Step3, otherwise end.

IV. Function Optimization Test and Analysis

In order to verify the optimization performance of DGA, the following five different functions are provided to test on a computer using Matlab[6]. The test results are compared with those of the SGA. In DGA, $\boldsymbol{N}$=20, f_a=0.5, f_b=1.5, k_{max}=1000, P_h=0.25. In SGA, N=20, P_c=0.3, P_m=0.1, k_{max}=1000. Considering the randomness of intelligent algorithms, each function was independently tested with 50 repetitions.

1. *Schaffer's* Function

$$\min\ f_1 = 0.5 + ((\sin(\sqrt{x_1^2 + x_2^2}))^2 - 0.5)/((1 + 0.001(x_1^2 + x_2^2))^2) \quad (5)$$

$x_1, x_2 \in [-10,10]$, f^*=0.

2. *Ackley*'s Path function

$$\min\ f_2 = -20\exp(-0.2\sqrt{0.5(x_1^2 + x_2^2)}) - \exp(0.5(\cos(2\pi x_1) + \cos(2\pi x_2))) + \exp(1) + 20 \quad (6)$$

$x_1, x_2 \in [-5.12,5.12]$, f^*=0.

3. *Rastrigrin* function

$$\min\ f_3 = 20 + x_1^2 - 10\cos(2\pi x_1) + x_2^2 - 10\cos(2\pi x_2) \quad (7)$$

$x_1, x_2 \in [-5.12,5.12]$, f^*=0.

4. *Michalewicz's* function

$$\min\ f_4 = -\sin(x_1)(\sin(x_1^2/\pi))^{20} - \sin(x_2)(\sin(2x_2^2/\pi))^{20} \quad (8)$$

$x_1, x_2 \in [0,\pi]$, f^*=-1.8013.

5. *Camel* function

$$\min\ f_5 = [4 - 2.1x_1^2 + x_1^4/3]x_1^2 + x_1x_2 + (-4 + 4x_2^2)x_2^2 \quad (9)$$

$x_1, x_2 \in$ [-5.12,5.12], f^*=-1.031628.

Table 1 Comparison results among three algorithms

f	ε	$\boldsymbol{N_{best}}$		$\boldsymbol{N_{max}}$		$\boldsymbol{N_{mean}}$		$\boldsymbol{N_{sd}}$	
		SGA	DGA	SGA	DGA	SGA	DGA	SGA	DGA
f_1	10^{-13}	49	50	706	564	355.98	286.54	145.25	86.66
f_2	10^{-11}	49	50	849	624	393.41	292.86	153.30	115.66
f_3	10^{-6}	48	50	557	398	311.35	221.14	103.78	77.55
f_4	10^{-3}	31	47	1000	773	434.97	291.98	256.54	216.14
f_5	10^{-3}	42	50	861	451	331.43	236.08	251.93	103.75

Table 1 is the comparison results among DGA and SGA. $\boldsymbol{N_{best}}$ denotes the number of times an optimal solution was found. $\boldsymbol{N_{max}}$, $\boldsymbol{N_{mean}}$ and $\boldsymbol{N_{sd}}$ denote the maximum, average convergence and standard deviation generations needed to find the optimal solutions, respectively. From the comparison results, we can see that the optimization results of DGA are better than those of SGA, which shows the strong optimization ability of the DGA. DGA improved the global search capability of SGA by adaptive differential evolution operator. Under the condition of ensuring population diversity, the algorithm performs adaptive differential evolution operator according to the fitness value after the individual fitness is tested. So the whole population quality is improved and the risk of local convergence is reduced. The global searching ability of DGA also is improved.

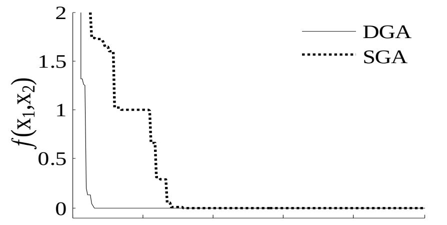

Fig. 2 Average evolutionary curves of optimal solution Of *Rastrigrin* function

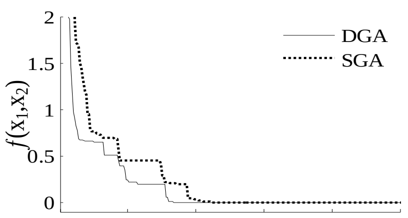

Fig. 3 Average evolutionary curves of whole population of *Rastrigrin* function

Fig.2 gives average evolutionary curves of optimal solution of *Rastrigrin* function. From the two curves, it can also be seen that the convergence speed of the proposed DGA is faster than the speed of SGA, which further verify the effectiveness of the DGA. Fig.3 gives Average evolutionary curves of whole population of *Rastrigrin* function. From the two curves, it can also be seen that the population convergence performance of the proposed DGA is better than that of SGA, and the search stability of DGA is higher than that of SGA.

V. Conclusions

In order to solve the problem of premature convergence and improve the optimization ability of the simple genetic algorithms during the complex function optimization, an improved genetic algorithm based on adaptive differential evolution is presented. Compared with the simple genetic algorithm, computer simulation shows that the convergence speed of the proposed DGA algorithm is faster and the global search ability is improved. Furthermore, the DGA can effectively reduce the dependence of the convergence on the initial population.

Acknowledgements

This work is supported by the 2015 College Students Practice Innovation Training Program of Jiangsu Province, 2015 Entry Project of Service Center of Zhangjiagang Camphor Tree Makerspace, and Modern Educational Technology Project of Jiangsu Province (No. 2015-R-40701).

References

[1]. L.B. Booker, D.E. Goldberg, J.H. Holland, Classifier systems and genetic algorithms, Artificial Intelligence, 89,1989,235–282.

[2]. G. Gong, G.S. Hao, F. Yang, An improved algorithm for the suppression of premature convergence of genetic algorithms, Computer and Digital Engineering, 5, 2009, 6-9, 16.

[3]. L. Ling, Y.J. Hu, Q.Q. Wang, J.J. Zhang. Optimization of hole group machining path based on improved genetic algorithm, Journal of Huazhong University of Science and Technology (NATURAL SCIENCE EDITION), 8, 2009, 88-91.

[4]. R. Storn, K. Price, Differential evolution-a simple and efficient heuristic for global optimization over continuous space, Journal of Global Optimization, 4, 1997, 341-359.

[5]. Y.L. Zhou, Y.H. Zhu, Discrete differential evolution with fitness uniform selection scheme, Journal of Chinese Computer Systems, 1, 2012, 151-154.

[6]. M.X. Yuan, H.F. Du, S.A. Wang, J. Zhuan, Chaos small-world optimal algorithm based on population entropy,Journal of Xi'an Jiaotong University,9,2008,1137-1141.