A New Adaptive Differential Evolution Algorithms

Yu Tian\textsuperscript{a}, Tinghui Li\textsuperscript{b}

College of Electronic Engineering, Guangxi Normal University, Guilin, China
\textsuperscript{a}575392501@qq.com, 8618607732871
\textsuperscript{b}tinghuili@gxnu.edu.cn, 8613977305876

ABSTRACT. In this paper, we describe a New Adaptive Differential Evolution algorithm (NADE) based on adaptive mutation operator, crossover operator and new mutation strategy. It is mainly aimed at the existence of individual aggregation and the reduction of population diversity in the calculation process of Differential Evolution algorithm (DE), which makes the algorithm easy to get early. The problems of ripeness, slow convergence speed and low convergence accuracy are improved. The improved differential evolution algorithm is tested by five commonly used test functions, and the test results are compared with the other three algorithms. The results show that the proposed algorithm performs better in convergence speed, convergence precision and global convergence ability.

1. INTRODUCTION

Differential Evolutionary Algorithms \cite{1} was proposed by Rainer Storn and Kenneth Price in 1995 on the basis of evolutionary ideas such as genetic algorithm. The basic idea of DE algorithm is derived from genetic algorithm. The original idea was to solve Chebyshev polynomial problems \cite{2}, which later became an effective computational technique for solving complex optimization problems. DE algorithm adopts the evolutionary strategy of mutation, crossover and selection \cite{3}. It has the characteristics of fast convergence, simple algorithm and strong robustness. The algorithm is widely used in electric power system \cite{4}, artificial neural network \cite{5}, electromagnetics \cite{6}, iris registration \cite{7} and other fields. However, in the later stage of evolution of DE algorithm, due to the reduction of the diversity among individuals, it is easy to appear problems such as search stagnation, premature convergence, convergence accuracy reduction and convergence speed decline \cite{8}.

In view of the above shortcomings, a large number of scholars at home and abroad have put forward many suggestions for improvement. Document \cite{9} proposes an adaptive differential evolution algorithm (DDE) which is considered to be one of the earliest. In document \cite{10}, a differential evolution algorithm (JDE) with adaptive control parameters is proposed. A new differential evolution algorithm based on adaptive mutation operator and crossover operator is proposed in document \cite{11}. In document \cite{12}, an adaptive differential evolution algorithm for selecting crossover operator adaptively is proposed.

In this paper, a new adaptive differential evolution algorithm (NADE) is proposed based on the analysis of the influence of evolution mode and control parameters on DE algorithm, and several standard functions are selected for testing. The results show that the algorithm has good global search ability and strong convergence.
2. STANDARD DE ALGORITHM

The basic idea of standard differential evolution algorithm is to mutate, cross and select the initial population generated randomly, and to achieve the survival of the fittest individuals through continuous iteration and updating. The main steps are as follows.

2.1 Population Initialization

In the initial stage of population, a series of basic parameters are determined according to practical problems, including Population Number, Spatial Dimension, Number of iterations, mutation operator F, crossover operator CR, search space upper \( x_{j,0}^{\max} \) and search space lower \( x_{j,0}^{\min} \). The initial population is randomly generated in \([x_{j,0}^{\min}, x_{j,0}^{\max}]\) \((i = 1,2,...,\text{NP}; j = 1,2,...,N)\). The expression for this operation is:

\[
x_{j,0}^0 = x_{j,0}^{\min} + \text{rand}(0,1) \cdot (x_{j,0}^{\max} - x_{j,0}^{\min})
\]

In this expression, \( x_{j,0}^0 \) represents the j-dimension of the i-th individual of the 0th generation, and \( \text{rand}(0,1) \) represents the random number which obeys uniform distribution in the open interval (0,1).

2.2 Mutation Operation

The basic principle of mutation operation is to sum the difference vectors and the base vectors to generate the mutation vectors. The most commonly used mutation strategy in mutation operation is DE/rand/1/bin. The expression for this operation is:

\[
v_i^{\theta+1} = x_{r_3}^\theta + F \cdot (x_{r_1}^\theta - x_{r_2}^\theta), \quad i \neq r_1 \neq r_2 \neq r_3
\]

In this expression, F is a mutation operator, \( i, r_1, r_2, r_3 \) are random integers that are different from each other in the group 1,2,...,NP, and \( v_i^{\theta+1} \) is the mutated individual, and \( x_i^\theta \) is the i-th individual of the g-th generation.

2.3 Crossover Operation

The basic principle of crossover operation is to produce test vectors by mixing parameters of target vectors and variation vectors. The expression for this operation is:

\[
u_{i,j}^{\theta+1} = \begin{cases} v_{j}^{\theta+1}, & \text{if } \text{rand}(0,1) \leq \text{CR} \text{ or } j = j_{\text{rand}} \\ x_{j}^{\theta}, & \text{otherwise} \end{cases}
\]

In this expression, CR is a crossover operator and \( j_{\text{rand}} \) is a random integer on the group 1,2,...,N. It is used to ensure that at least one of the variant individuals \( v_i^{\theta+1} \) is passed on to the next generation, and the rest are crossed.

2.4 Selection Operation

The basic principle of the selection operation is that if the fitness value of the test vector is better than that of the target vector, then the test vector is used to replace the target vector in the next generation, otherwise the target vector is saved, that is, individuals with better fitness are selected to enter the next generation based on the greedy strategy principle. The expression for this operation is:

\[
x_{j}^{\theta+1} = \begin{cases} u_{j,i}^{\theta+1}, & \text{if } f(u_{j,i}^{\theta+1}) \leq f(x_{j}^{\theta}) \\ x_{j}^{\theta}, & \text{otherwise} \end{cases}
\]

2.5 Operation Aborted

When the fitness value of an individual in \( x_{j}^{\theta+1} \) satisfies the end condition of the algorithm, the algorithm ends and outputs the individual; otherwise, step 2.2 is returned to continue mutation, crossover and selection operations.
3. NADE ALGORITHM

The new adaptive differential evolution algorithm (NADE) optimizes DE algorithm mainly from the selection of mutation operator F, cross operator CR and the improvement of mutation strategy.

3.1 Optimization of Mutation Operators

In the mutation operation, the product of the difference vector of two parent individuals and the mutation operator F is taken as the mutation step, and then the mutation step is added to the third parent individual, and the mutation step is screened to produce a new offspring individual. Formula (2) shows that the value of mutation operator F has a great influence on the convergence and convergence speed of the algorithm. When the value of F is larger, the wider the search range of the algorithm is, the stronger the global search ability is, which is conducive to convergence to the global optimal solution, but the convergence speed is slower; when the value of F is smaller, the convergence speed of the algorithm is faster and the local search ability is stronger.

In the standard DE algorithm, the mutation operator F is constant, which cannot take into account the global search ability and convergence speed. According to the previous analysis, in order to make the algorithm have better global search ability and convergence speed, the value of mutation operator F needs to be larger in the early stage, and then gradually decreases. Based on the above principles, a new adaptive mutation operator is proposed in NADE algorithm. The expression for this operation is:

\[ F = F_0 \cdot 2^\lambda \]

\[ \lambda = \exp \left( 1 - \frac{G_m}{G_m+1} \right) \]

In these two formulas, \( F_0 \) denotes the initial value of the mutation operator, \( F_0 = 0.9 \), \( G_m \) denotes the maximum evolutionary algebra, and \( G \) denotes the current iteration number. From formula (5) and formula (6), we can see that the variation operator F has a linear decreasing trend.

3.2 Optimization of Mutation Strategy

The selection of mutation strategy is an important factor to determine the global and local optimization of the algorithm. Different mutation strategies have an important impact on the search results. Formula (2) is the most basic and commonly used mutation strategy, but at the later stage of the search, it often falls into the local optimum, which makes the convergence speed greatly reduced and prone to premature convergence. On the basis of formula (2), this paper adds a random perturbation to the two parents who perform differential operation, and replaces the original fixed step size with the random step size. The expression for this operation is:

\[ v_i^r+1 = x_{r3}^r + F \cdot (\text{rand} \cdot x_{r1}^r - \text{randn} \cdot x_{r2}^r), \]

\[ i \neq r1 \neq r2 \neq r3 \]

In this formula, \( \text{rand} \) is a random number between zero and one, \( \text{randn} \) is a normal distribution random number with mean value of zero and variance of one.

3.3 Optimization of Crossover Operator

The size of the crossover operator CR determines the probability that the new individual will inherit the "gene" from the mutant individual or the parent individual. Formula (3) shows that when the value of CR is large, the probability of new individuals inheriting "gene" from the mutant individuals is higher, which is beneficial to enhancing the local search ability of the algorithm; when the value of CR is small, the probability of new individuals inheriting "gene" from the parent individuals is higher, which is beneficial to maintaining the diversity of the population.

In the standard DE algorithm, the crossover operator CR takes a constant value, which cannot take into account the global search ability and search speed. According to the previous analysis, in order to prevent the algorithm from falling into local optimum, it is necessary to maintain the diversity of the population. In order to accelerate the convergence speed, it is necessary to enhance the local convergence ability of the algorithm. The value of CR needs to be smaller in the early stage, and then
gradually increase. In view of this feature, a new adaptive crossover operator is proposed in NADE algorithm. The expression for this operation is:

\[ CR = CR_0 \cdot (1 - F) \]

In this formula, \( CR_0 \) represents the initial value of the crossover operator, and \( CR_0 = 0.8 \). The expressions of the adaptive mutation operator \( F \) are shown in Formula (5) and Formula (6). Formula (8) shows that the value of crossover operator \( CR \) is opposite to that of mutation operator \( F \), and the value of crossover operator \( CR \) increases monotonously.

4. ALGORITHMIC TESTING

In order to verify the effectiveness of the proposed algorithm, five standard test functions are selected from reference [13-14], and compared with standard DE algorithm, JDE algorithm and DDE algorithm. Five standard test functions are listed in Table 1.

Table 1 Standard test function

| Function | Expression | Range | Global Optimal Solution |
|----------|------------|-------|-------------------------|
| Sphere   | \( f_1(x) = \sum_{i=1}^{D} x_i^2 \) | [-100, 100] | 0 |
| Rastrigin | \( f_2(x) = \sum_{i=1}^{D} [x_i^2 - 10 \cos(2\pi x_i) + 10] \) | [-5.12, 5.12] | 0 |
| Griewank | \( f_3(x) = \sum_{i=1}^{D} \frac{x_i^2}{4000} - \prod_{i=1}^{D} \cos \left( \frac{x_i}{\sqrt{i}} \right) + 1 \) | [-600, 600] | 0 |
| Schwefel’s 2.22 | \( f_4(x) = \sum_{i=1}^{D} |x_i| + \prod_{i=1}^{D} |x_i| \) | [-10, 10] | 0 |
| Ackley   | \( f_5(x) = -20 \exp \left( -0.2 \sqrt{\frac{1}{D} \sum_{i=1}^{D} x_i^2} \right) - \exp \left( \frac{1}{D} \sum_{i=1}^{D} \cos 2\pi x_i \right) + 2 + e \) | [-32, 32] | 0 |

The simulation experiment is programmed with the software of MATLAB R2018a, and the experiment is configured as Inter (R) Core (TM) i3-7100 CPU @ 3.90 GHz. The basic parameters of the algorithm are as follows: dimension \( D = 30 \), population size \( NP = 50 \); the parameter setting of JDE algorithm is shown in reference [10]; the parameter setting of DDE algorithm is shown in reference [9]; In NADE algorithm, \( F_0 = 0.9, \ CR_0 = 0.8 \), and other parameters are consistent with the above algorithms. Table 2 shows the results of 30 times running of five test functions independently and taking average values. The results mainly include: optimal value, average value, standard deviation, convergence time and winning rate.

Table 2 The results function optimization

| function | algorithm | optimal value | average value | standard deviation | Convergence time /s | Success rate /% |
|----------|-----------|---------------|---------------|--------------------|---------------------|-----------------|
| Sphere   | DE        | 2.66E-014     | 1.46E-009     | 2.17E-009          | 10.77               | 100             |
|          | JDE       | 5.52E-025     | 1.15E-023     | 3.34E-022          | 5.19                | 100             |
|          | DDE       | 2.91E-043     | 3.00E-034     | 1.19E-032          | 5.30                | 100             |
|          | NADE      | 8.31E-106     | 1.50E-105     | 3.96E-103          | 10.76               | 100             |
| Rastrigin| DE        | 5.95E+000     | 14.56E+000    | 5.02E-001          | 11.38               | 100             |
|          | JDE       | 1.36E-005     | 7.51E-002     | 3.44E-001          | 5.18                | 100             |
By analyzing Table 2, we can see that when the dimension is 30, NADE algorithm is better than standard DE algorithm, JDE algorithm and DDE algorithm in five aspects: optimum value, average value, standard deviation, convergence time and success rate. The optimal value shows the effectiveness of an algorithm, while the average value, standard deviation and success rate show the optimization ability and stability of the algorithm from another angle. In summary, NADE algorithm is superior to other three algorithms in convergence accuracy and global search ability.

5. CONCLUSIONS

Based on the analysis of the factors affecting the optimization effect of DE algorithm, this paper proposes an adaptive mutation operator, an adaptive crossover operator and an improved mutation strategy, which makes the improved algorithm take into account the ability of global search and local development. Five standard test functions are used to optimize the calculation. The experimental results show that the NADE algorithm has faster convergence speed, higher convergence accuracy and better global convergence ability than other algorithms. However, this method still has some shortcomings. It can be seen from Table 2 that the running time of NADE algorithm is longer when some functions are tested. In the future, the influence of simultaneous multi-population evolution and parameter adaptive selection on DE algorithm will be further studied.

REFERENCES

[1] Storn, R. and Price, K. 1997. Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces. Journal of Global Optimization, 11(4), 341-359. DOI= 10.1023/a:1008202821328.

[2] Yuhong, H. 2012. Chebyshev inequality and its application. Journal of Changchun University of Technology, 33 (06), 712-714. DOI= 10.3969/j.issn.1674-1374.2012.06.024.

[3] Dexuan, Z. and Liquan, G. 2012. An efficient improved differential evolution algorithm. In The 31st Chinese Control Conference (Hefei, China, July 01, 2012), 2385-2390.

[4] Chengfu, S., Jianyang, Zh. and Lei, G. 2017. Cascade hydrothermal power system scheduling based on variable weight factor differential evolution algorithm. Data acquisition and processing, 32(01), 95-103. DOI= 10.16337/j.1004-9037.2017.01.011.

[5] Mu, L., Yigang, H., Shaowu, Zh and Zurun, L. 2010. A Differential Evolution Algorithms for Optimizing Wavelet Neural Networks and Its Application in Weak Signal Detection.
6

[6] Jing, S., Jun, Zh., Hua, L. and Canxia, W. 2019. Application of resonant radio power transmission technology in UAV. *Electronic Science and Technology*, 32 (04), 49-53.

[7] Dexuan, Z., Xin, W. and Na, D. 2013. An Iris Location Algorithm Based on Modified Differential Evolution. *Control Theory and Applications*, 30 (09), 1194-1200. DOI= 10.7641/CTA.2013.12223.

[8] Liang, C. 2012. Improved adaptive differential evolution algorithm and its application. Doctoral Thesis. *Donghua University*. DOI= 10.1016/j.chemolab.2013.07.004.

[9] Price, K. V., Storn, R. M. and Lampinen, J. A. 2005. Differential evolution - a practical approach to global optimization. *Natural Computing*, 141(2). DOI= 10.1007/3-540-31306-0.

[10] Brest, J., Greiner, S. and Boskovic, B. 2006. Self-Adapting Control Parameters in Differential Evolution: A Comparative Study on Numerical Benchmark Problems. *IEEE Transactions on Evolutionary Computation*, 10(6), 646-657. DOI= 10.1109/tevc.2006.872133.

[11] Longlong, L. and Qisheng, Y. 2017. A new improved differential evolution algorithm. *Jiangxi Science*, 35 (04), 485-489. DOI= 10.13990/j. issn1001-3679.2017.04.001.

[12] Jingting, X. 2010. Research and Application of Improved Differential Evolution Algorithms. *Electronic Technology*, 47(5). DOI= 10.3969/j.issn.1000-0755.2010.05.006.

[13] Chakraborty, U. K., Rahnamayan, S., Tizhoosh, H. R. and Salama, M. M. A. 2008. Opposition-based differential evolution. DOI= 10.1007/978-3-540-68830-3_6.

[14] Ali, M. M., Khompatraporn, C. and Zabinsky, Z. B. 2005. A numerical evaluation of several stochastic algorithms on selected continuous global optimization test problems. *Journal of Global Optimization*, 31(4), 635-672. DOI= 10.1007/s10898-004-9972-2.