A Differential Evolution Algorithm with Adaptive Population Size Reduction Strategy

Xiaoyan Zhang(✉), Zhengyu Duan, and Qianqian Liu

Xi'an University of Science and Technology, Xi'an 710000, China
zhangxy@xust.edu.cn

Abstract. Aiming at the problem that the differential evolution algorithm easily falls into a local optimum and results in premature convergence, a new differential evolution algorithm with an adaptive population size reduction strategy (APRDE) is proposed. Firstly, in the mutation and crossover operation, to balance the local exploitation and global exploration capabilities of the algorithm, a parameter adaptive tunning scheme based on the hyperbolic tangent function and Cauchy distribution is proposed to adaptively adjust the parameter factors. Secondly, an ordered mutation strategy is adopted to guide the direction of mutating and enrich the diversity of the population. Lastly, after each evolution iteration, adaptively reducing the population size according to the error between the fitness values of individuals and the current optimal. The proposed algorithm is compared with 5 other optimization algorithms on 8 typical benchmark functions. The results show that the algorithm has a great improvement in solution accuracy, stability and convergence speed.

Keywords: Adaptive differential evolution algorithm · Parameters tunning scheme · Ordered mutation strategy · Population size reduction strategy

1 Introduction

Coal has always been the main energy source in China, and accounts for more than 60% of consumption, it will remain be the main energy source in China until 2050. Coal intelligent mining is a new stage in the development of coal comprehensive mining technology, which is also an inevitable requirement for the technological revolution and upgrading development of coal industry [1]. Three-dimensional modeling of coal seams at the fully mechanized mining face is an important foundation for coal enterprises to realize "intelligent management and transparent mining".

Researchers often use kriging interpolation to interpolate unknown regions in space to build 3D models. For the problem that kriging interpolation is prone to overfitting or underfitting in the fitting process of variogram, this paper proposes a Differential Evolution algorithm with Adaptive Population size Reduction (APRDE) to optimize the kriging interpolation algorithm. After experimental verification, the adaptive differential evolution algorithm proposed in this paper has higher solution accuracy, faster convergence speed and better stability.

2 The APRDE Algorithm

The Differential Evolution (DE) algorithm [2] is an effective heuristic search algorithm that can be used to solve parameter optimization problems. This paper optimizes the DE algorithm by modifying the variation strategy, parameter adaptive adjustment mechanism, and population reduction strategy.

2.1 Variation Strategy

The mutation process and the crossover process are the core parts of the differential evolution algorithm, and the mutation formula is:

$$v_i = x_i + F\left(x_j - x_r\right) \tag{1}$$

where v_i is the individual after mutation, x_i, x_j, x_r are the random individuals in the current population and $i \neq j \neq r$, In the process of mutation, in order to enrich the diversity of the population and improve the local exploitation ability of the algorithm, this paper adopts an ordered mutation strategy [3], and the mutation equation is:

$$v_i = x_i + F(x_{best} - x_i) + F(x_{middle} - x_{worst}) \tag{2}$$

Three randomly selected individuals from the current population are sorted according to the fitness value to obtain $x_{best}, x_{middle}, x_{worst}$. With the current vector as the base vector, avoiding the algorithm from falling into a local optimal solution or stagnation. Combining the base vector and two ordered difference vectors, enriching the diversity of the population, and making the direction of variation gradually approach the optimal solution.

2.2 Parameter Adaptive Adjustment

In the variation process, the variation factor F controls the magnitude of the base vector change. To balance the global exploration and local exploitation abilities of the algorithm, the hyperbolic tangent curve between [−4,4] [4] is used in this paper to control the variation of the mutation factor with the following variation equation.

$$F = \frac{F_{max}+F_{min}}{2} + \frac{\tanh\left(-4+8\frac{G_{max}-G}{G_{max}}\right)(F_{max}-F_{min})}{2} \tag{3}$$

where $F_{\min}$ is the minimum value and $F_{\max}$ is the maximum value of the variation factor. G_{max} is the maximum evolutionary generation and G is the current evolutionary generation. The hyperbolic tangent curve changes very little at the beginning and the end. The variation factor varies approximately linearly between the maximum and minimum values, striking a balance between global exploration and local exploitation ability. Besides, a variation factor based on normal distribution is used in this paper to enhance the diversity of the variation vector and jump out of the local optimal solution. The variation process of the variation factor is as follows:

$$F = \begin{cases} randn(0.5, 0.1), & \left|x_{best,g} - x_{best,g+1}\right| < 10^{-8} \\ equation(7), & otherwises \end{cases} \tag{4}$$

The crossover process of the differential evolution algorithm is as follows:

$$u_{i,j} = \begin{cases} v_{i,j}, & rand(0, 1) < CR \; or \; j = rand(1, D) \\ x_{i,j}, & otherwises \end{cases} \tag{5}$$

The crossover process is to operate on each variable in an individual, and D denotes the number of variables. The variables in the mutated individual are crossed with those in the initial individual by setting the conditions to obtain the crossover individual. To accommodate the crossover process, this paper changes the crossover factor in a linearly reduced manner with the evolutionary process, and the change equation is:

$$CR = CR_{max} - \frac{G(CR_{max} - CR_{min})}{G_m} \tag{6}$$

where the change range of the crossover factor is $[CR_{min}, \; CR_{max}]$. The adaptive differential evolution algorithm proposed in this paper adaptively changes the variance and crossover factors in evolutionary process, and balances the global exploration and local exploitation ability to some extent in the algorithm search process.

2.3 Population Reduction Strategy

Reduction of populations during evolution process of differential evolution algorithm can effectively capture useful individual information, reduce unnecessary computational resources, and improve convergence speed. This paper proposes a nonlinear population reduction strategy to control the reduction of population size according to the hyperbolic tangent function curve between [−2.5,4]. Through a predetermined maximum evolutionary generation $G_{\max}$, the reduction function changes with the current evolutionary generation as the independent variable. The reduction equation is as follows:

$$F = \frac{NP_{max}+NP_{min}}{2} + \frac{\tanh\left(-2.5+4\cdot\frac{G_{max}-G}{G_{max}}\right)(NP_{max}-NP_{min})}{2} \tag{7}$$

In the early stage of evolution, the size of population is large, and to ensure the diversity of population and to improve the global search ability of the algorithm. As the evolutionary process proceeds, the solved optimal individuals are closer and closer to the optimal solution. To save computational resources and improve the convergence speed, some individuals far from the optimal solution are removed. In the later stage of the evolutionary process, the population iterates around the optimal solution. To improve the local search ability, the population size is maintained at the small value and local search is performed carefully to ensure that the optimal solution in that range is found.

3 Numerical Experiment and Analysis

The experiments in this paper use a 64-bit Windows 10 operating system. The processor is an Intel(R) Core (TM) i5-5200U CPU @ 2.20 GHz with an Intel(R) HD Graphics 5500 GPU. Python 3.5.2 is selected as the experimental code language, and the experiment is run in PyCharm software to complete the experimental process.

3.1 Experiments Setup

In this paper, we choose five comparison algorithms, namely DE [2], LSHADE (Linear Success-History based Adaptive DE) [5], AGDE (Adaptive Guided Differential Evolution) [6], AMODE (a DE algorithm based on Adaptive Mutation Operator) [7] and ASVDE (modified DE algorithm based Adaptive Secondary Variation) [8]. The comparative analysis of the algorithms is performed on eight typical benchmark functions in CEC2014 [9], which are unimodal f_1, f_2, simple multimodal f_6, f_{12}, hybrid f_{17}, f_{22}, and composite functions f_{24}, f_{27}, as shown in Table 1. D is the dimensionality of the problem. The search space of these benchmark functions is [-100,100], with more local optima and function values greater than 0. Therefore, the fitness function is defined as f $= f(x) - f(x^*)$. $f(x)$ is the function value calculated by the algorithm, $f(x^*)$ is the known optimal value of the function. The closer the f is to zero, the closer the function value calculated by the algorithm is to the global optimum. The parameter variables set for the comparison experiments are shown in Table 2.

Table 1. Some benchmark functions of CEC2014

f	No.	Functions	$F_i^* = F_i(x^*)$
f_1	1	Rotated High Conditioned Elliptic Function	100
f_2	2	Rotated Bent Cigar Function	200
f_3	6	Shifted and Rotated Weierstrass Function	600
f_4	12	Shifted and Rotated Katsuura Function	1200
f_5	17	Hybrid Function 1 (N = 3)	1700
f_6	22	Hybrid Function 6 (N = 5)	2200
f_7	24	Composition Function 2 (N = 3)	2400
f_8	27	Composition Function 5 (N = 5)	2700

3.2 Comparison of Solution Accuracy and Stability

With the variable settings and experiments conducted in Table 2, the six comparison algorithms were run 21 times on the benchmark functions in the dimensions of D = 30., the average (avg) and standard deviation (std) of fitness function values were calculated and recorded in Tables 3, with the optimal values bolded in the table.

As shown in Table 3, the mean values solved by the APRDE algorithm on eight functions are 5.51E+02, 0.00E-00, 2.65E-01, 6.00E-01, 2.24E+02, 2.84E+01, 2.22E+02 and 3.20E+02, which are closer to the optimal solution than the mean values solved by other algorithms. The results obtained by the APRDE algorithm are closer to the optimal solutions compared with other algorithms, and also have better performance for solving high-dimensional optimization problems. The APRDE algorithm solves to the closest global optimal solution on 88% of the functions. The standard deviation range obtained by APRDE is 0.00E+00 to 3.85E+05, which has the smallest value compared with other algorithms, so the algorithm has the best stability.

Table 2. Parameter settings of comparison algorithms

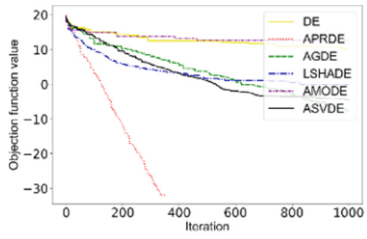
DE	APRDE	AGDE	LSHADE	AMODE	ASVDE
$G_{\max} = D * 100$ $NP = 100$					
$F = 0.75$	$F_{\max} = 0.9$	$F = randn(0.1, 1)$	$F = 0.75$	$F = 0.75$	$F_{\max} = 0.9$
$CR = 0.7$	$F_{\min} = 0.2$	$F_{\max} = 0.2$	$CR = 0.7$	$CR = 0.7$	$F_{\max} = 0.2$
	$CR_{\max} = 0.9$	$CR_2 = [0.9, 1.00]$	$p = 0.1$	$C = [0.05, 0.95]$	$Mr = 0.99$
	$CR_{\min} = 0.2$		$H = 5$		$Max\_count = 5$
	$NP_{\min} = 50$		$MF = MCR = 0.5NP_{\min} = 10$		$CR = 0.7$

Table 3. Results of comparison algorithms on benchmark functions for D = 30

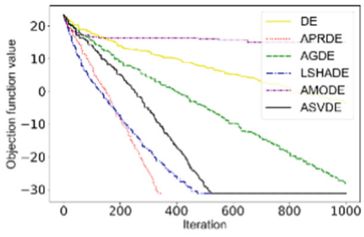
Function	Criterion	DE	AGDE	LSHADE	AMODE	ASVDE	APRDE
f_1	avg	5.46E + 07	1.97E+04	1.43E+04	2.91E+05	1.11E+06	**5.51E+02**
	std	1.57E+07	3.91E+03	3.45E+03	1.03E+05	1.05E+06	**3.91E+02**
f_2	avg	2.73E+07	9.86E+03	7.15E+03	1.46E+05	5.53E+05	**0.00E+00**
	std	2.95E+07	1.02E+04	7.55E+03	1.63E+05	9.27E+05	**0.00E+00**
f_3	avg	1.82E+07	6.57E+03	4.77E+03	9.70E+04	3.69E+05	**2.65E-01**
	std	2.73E+07	9.57E+03	7.03E+03	1.49E+05	8.00E+05	**5.29E-01**
f_4	avg	1.36E+07	4.93E+03	3.58E+03	7.28E+04	2.77E+05	**6.00E-01**
	std	2.49E+07	8.76E+03	6.43E+03	1.36E+05	7.11E+05	**8.26E-02**
f_5	avg	1.10E+07	4.03E+03	3.11E+03	6.11E+04	2.22E+05	**2.24E+02**
	std	2.29E+07	8.04E+03	5.82E+03	1.24E+05	6.46E+05	**1.44E+02**
f_6	avg	9.16E+06	3.37E+03	2.61E+03	5.10E+04	1.85E+05	**2.84E+01**
	std	2.13E+07	7.49E+03	5.43E+03	1.15E+05	5.95E+05	**6.82E+00**
f_7	avg	7.86E+06	2.92E+03	2.27E+03	4.37E+04	1.58E+05	**2.22E+02**
	std	2.00E+07	7.02E+03	5.10E+03	1.08E+05	5.55E+05	**2.65E-01**
f_8	avg	6.87E+06	2.60E+03	2.03E+03	3.83E+04	1.39E+05	**3.20E+02**
	std	1.89E+07	6.62E+03	4.81E+03	1.02E+05	5.22E+05	**4.02E+01**

3.3 Comparison of Convergence Speed

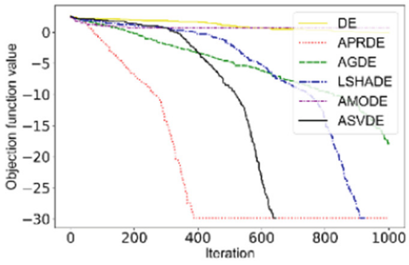
When D = 10, the convergence curves of the six compared algorithms on each function are shown in Fig. 1. From Figs. 1-a and 1-b, all algorithms show a single decreasing trend in the process of solving for unimodal functions, and the solution results are near the optimal solution. Compared with other algorithms, the APRDE algorithm converges the fastest, finds the global optimal solution first, and has the highest solution accuracy. In the simple multimodal functions f_3 and f_4, the fitness function values solved by



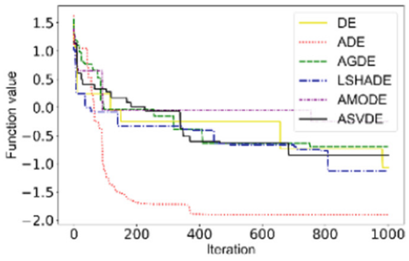
(a) f_1



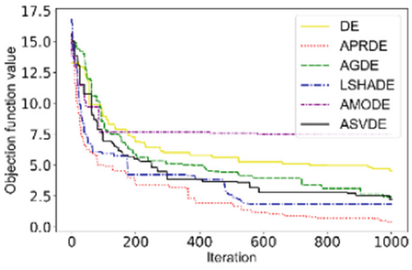
(b) f_2



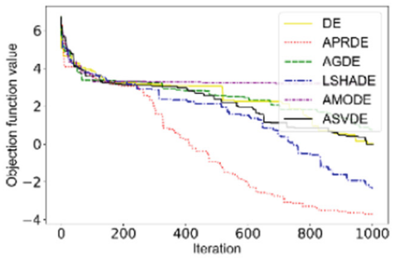
(c) f_3



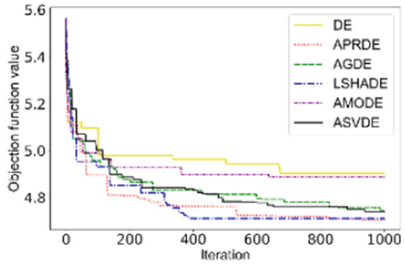
(d) f_4



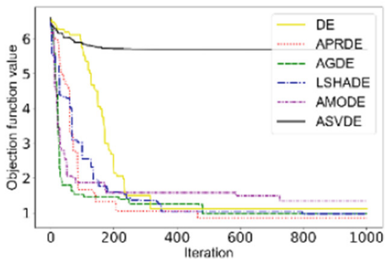
(e) f_5



(f) f_6



(e) f_7



(f) f_8

Fig. 1. The convergent curve of comparison algorithms on eight benchmark functions.

APRDE algorithm keep decreasing in the iterative process, the algorithm converges faster and first finds the global optimal solution first in the f_3 function, and in the f_4 function, the solved optimal is closer to the global optimal solution. Among the above four functions, compared with the ASVDE, LSHADE and AGDE algorithms, the APRDE algorithm not only finds the global optimal solution, but also converges faster and more efficiently. In the hybrid and composite functions $f_5 \sim f_8$, the functions have multiple local optimal solutions and larger local optimal values. The comparison results shows that the convergence curve of the APRDE algorithm is decreasing and the solved results are smaller and closer to the global optimal solution.

4 Conclusion

In the process of solving optimization problems, the differential evolution algorithm tends to converge prematurely and falls into a local optimal solution or stagnation state. In this paper, we propose a Differential Evolution Algorithm with Adaptive Population size Reduction (APRDE) strategy to remedy the shortcomings of traditional differential evolution algorithm. The APRDE algorithm proposes a parameter adaptive adjustment mechanism to balance the global exploration and local exploitation ability in the search process and adopts an ordered variation strategy to enrich the diversity of the population and improve the convergence speed and solution accuracy of the algorithm. In the evolutionary process, a nonlinear population reduction strategy is proposed to save computational cost and improve the quality of computational results at the same time. Compared with the other five optimization algorithms, the APRDE algorithm proposed in this paper has a fast convergence speed, the optimal solution is obtained in 88% of the tested functions, and the stability of the algorithm is relatively high.

References

1. Wang, G.F., Zhang, D.S.: Innovation practice and development prospect of intelligent fully mechanized technology for coal mining. J. China Univ. Min. Technol. **47**(3), 459–467 (2018)
2. Storn, R., Price, K.: Differential evolution—a simple and efficient adaptive scheme for global optimization over continuous spaces J. Global Optim. **11**(4), 341–359 (1997)
3. Mohamed, A.W., Hadi, A.A., Jambi, K.M.: Novel mutation strategy for enhancing SHADE and LSHADE algorithms for global numerical optimization. Swarm Evol. Comput. **50** 100455 (2019)
4. Yan, Q.M., Ma, R.Q., Ma, Y.X.: Adaptive simulated annealing particle swarm optimization algorithm Journal of Xidian University **48**(04), 120–127 (2021)
5. Mohamed, A.W., Mohamed, A.K.: Adaptive guided differential evolution algorithm with novel mutation for numerical optimization. Int. J. Mach. Learn. Cybern. **10**(2), 253–277 (2017)
6. Tanabe, R., Fukunaga, A.S.: Improving the search performance of SHADE using linear population size reduction. In: 2014 IEEE congress on evolutionary computation (CEC), pp. 1658–1665. IEEE (2014)
7. Liao, X., Li, J., Luo, Y.: Differential evolution algorithm based on adaptive mutation operator Comput. Eng. Appl. **54**(6), 128–134 (2018)

8. Hu, F., Dong, Q., Lv, L.: Modified differential evolution algorithm based on adaptive secondary variation and its application Comput. Eng. Appl. **38**(7), 271–280 (2021)
9. Liang, J.J., Qu, B.Y., Suganthan, P.N.: Problem definitions and evaluation criteria for the CEC 2014 special session and competition on single objective real-parameter numerical optimization. Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China and Technical Report, Nanyang Technological University, Singapore, 635: 490 (2013)