ISSN: 2664-9640

Improved differential evolution algorithm

Binrui Li

School of Sichuan business college, Chengdu 611730, China;

876494902@qq.com

Abstract

Optimization model is a NP hard problem. This paper designs a heuristic algorithm for the solution of the model based on the improved differential evolution algorithm. Aiming at the shortcomings of traditional differential evolution algorithm, such as easy to fall into local optimum, this paper adjusts the fixed scaling factor F and probability crossover factor CR into adaptive parameters. Then, the results of the improved algorithm are compared and analyzed to prove the feasibility of the improved algorithm.

Keywords

Optimization model; Differential Evolution algorithm; Crossover Factor CR.

1. Introduction

Differential evolution (DE) is a stochastic heuristic algorithm first proposed by storn R and price k in 1995 to solve Chebyshev polynomial problems

SADE proposed in reference [1] improves the convergence rate of solving unconstrained optimization problems. References [2, 4] improve the scaling factor F and crossover probability factor Cr in de algorithm, but they do not use these improved de algorithms to solve bilevel programming problems. Reference [3] used De to solve bilevel programming problem, but did not improve its mutation strategy (De / current-to-rand / 1), nor did it optimize the parameters of the algorithm. In this paper, after adaptive improvement of scaling factor F and probability crossover factor Cr, the improved DE algorithm is used to solve the nonlinear integer bilevel programming problem of oilfield development planning.

2. Improved differential evolution algorithm

In the traditional DE algorithm, population size NP, scaling factor F and crossover probability factor CR will affect the convergence speed and accuracy of the algorithm. So in the specific problem, the three parameters should be selected.

2.1. Population size NP

In solving optimization problems, we must first set up the population size, and the population size of different problems is different, neither too large nor too small. If the population size is too large, the convergence speed will be greatly reduced, which will affect the performance of the algorithm. If the population size is too small, it will improve the convergence speed, but the diversity of the population is difficult to guarantee, and it is easy to fall into the local minimum, and the satisfactory results cannot be obtained. Many literatures show that the population selection is generally 5-10 times of the dimension of independent variable. For different problems, the population size suitable for the problem is selected. In special problems, if the result precision is pursued, the dimension should be more than 10 times, if the convergence speed is pursued, the dimension is less than 5 times, but the population size cannot be less than 4, otherwise, the mutation operation cannot be performed (because the random selection is

ISSN: 2664-9640

conducted in the mutation operation Three individuals, population size less than 4, is not random selection).

2.2. Scaling factor F.

The scaling factor F is a multiple of the difference between two random individuals in mutation operation, and it is an important factor to determine the richness of population diversity and the convergence speed of the algorithm. Many experiments show that the value range of F is [0,2], generally 0.5. When the value of F is large, it can be seen from figure 2-2 that the variation results in the feasible region change greatly, and the population diversity increases, which can effectively avoid the algorithm falling into local minimum, but the convergence speed will be affected. On the contrary, when f value is small, the range of mutation results in the feasible region becomes smaller, and the diversity of population is weakened, which makes the algorithm fall into local minimum and the result is premature. Therefore, it is necessary to choose different values of F for specific optimization problems. Experience shows that the value range of F is [0,1]. When the problem of multi peak is encountered, it is suggested to increase the value of F and NP in order to get the global optimal solution.

2.3. Cross probability factor CR.

The crossover probability factor (CR) controls the proportion of individuals produced by crossover operation in their parents, which is similar to gene mutation in genetics. Many experiments show that the value range of F is [0,1]. When Cr is small, the proportion of parent individuals is large, the mutation point is less, the convergence speed will be faster, but the diversity of the population can not be guaranteed, and it is easy to fall into local minimum; on the contrary, it is conducive to maintaining the diversity of the population and global search, and the result is generally the global optimal solution. Among the three parameters, the scaling factor f has the most obvious influence on the convergence speed and accuracy of the algorithm. In many references [1-4], the improvement of F is also in many aspects, and the basic idea of this paper is that when the iteration starts, the F is adjusted to a large extent, which can increase the population diversity and avoid premature. When iterating to the middle and late stage, the F is adjusted smaller to improve the convergence speed, so that in the whole optimization process, the convergence speed and accuracy of the algorithm are guaranteed. The improvement idea of Cr is similar to that of F, which will not be repeated here.

In reference [5], the author changed the mutation operation to.

$$V_{i,t} = W_{i,t} X_{r1,t} + K_{i,t}(X_{best} - X_{i,t}) + F_{i,t}(X_{pbest} - X_{i,t}) \quad (1)$$

Among them, $X_{r1,t}$ is the randomly generated individual, $X_{best,t}$ is the contemporary optimal individual in the iterative process, and $X_{pbesti,t}$ is the historical optimal individual in the iterative process, including all basic individuals $X_i(\tau)$, variant individuals $K_{i,t}$, and experimental individuals $F_{i,t}$.

In equation (1) $W_{i,t}$, $K_{i,t}$ and $F_{i,t}$ are the control parameters of mutation operation, which are improved to adaptive form by the author. In equation (2), the form of $W_{i,t}$, $K_{i,t}$ and $F_{i,t}$ are described, and the adaptive form of CR is also described.

ISSN: 2664-9640

$$\begin{cases} W_{i,t} = W_{\min} + (W_{\max} - W_{\min})\left(\left(2 - \exp\left(\frac{t}{Gen} In(2)\right)\right) \times \frac{1}{2} + \frac{f_{i,t} - f_{\min,t}}{f_{\max,t} - f_{\min,t}} \times \frac{1}{2}\right) \\ K_{i,t} = K_{\min} + (K_{\max} - K_{\min})\left(\left(2 - \exp\left(\frac{t}{Gen} In(2)\right) - 1\right) \times \frac{1}{2} + \frac{f_{i,t} - f_{\min,t}}{f_{\max,t} - f_{\min,t}} \times \frac{1}{2}\right) \\ F_{i,t} = F_{\min} + (F_{\max} - F_{\min})\left(\left(2 - \exp\left(\frac{t}{Gen} In(2)\right)\right) \times \frac{1}{2} + \frac{f_{\max,t} - f_{i,t}}{f_{\max,t} - f_{\min,t}} \times \frac{1}{2}\right) \\ CR_{i,t} = CR_{\min} + (CR_{\max} - CR_{\min})\left(\left(2 - \exp\left(\frac{t}{Gen} In(2)\right)\right) \times \frac{1}{2} + \frac{f_{i,t} - f_{\min,t}}{f_{\max,t} - f_{\min,t}} \times \frac{1}{2}\right) \end{cases} \quad (2)$$

In the above formula $w_0 = 0.1$, $w_{\min} = 0.2$, $w_{\max} = 0.9$, $F_{\min} = 0.3$, $F_{\max} = 0.9$, $CR_{\min} = 0.1$, $CR_{\max} = 0.9$, i is the number of iterations, NP is the population size, $f_{best,t}$ is the best individual and $f_{worst,t}$ is the worst individual.

3. Comparison and analysis of the results of general examples of the improved algorithm

In order to test the performance of the improved adaptive scaling factor F and crossover probability factor Cr in the improved algorithm, the performance comparison experiments are carried out on nine test functions by combining the classification strategy de / RAND / 1 and SDE. The number of trials was 300 and the population size was 200. Table 5 shows the value ranges and theoretical optimal solutions of nine commonly used test functions and independent variables.

Table 1 Standard function test set

Test function	Value range	Theoretical optimal solution
Sphere function $f_1(\vec{X}) = \sum_{i=1}^{D} x_i^2$	$-30 \le x_i \le 30$	$f_1(\bar{0}) = 0$
Schwefel's problem 1.2 $f_2(\vec{X}) = \sum_{i=1}^{D}\left(\sum_{j=1}^{i} x_j\right)^2$	$-30 \le x_i \le 30$	$f_1(\bar{0}) = 0$
High conditioned Elliptic function $f_3(\vec{X}) = \sum_{i=1}^{D}\left(10^6\right)^{\frac{i-1}{D-1}} x_i^2$	$-30 \le x_i \le 30$	$f_1(\bar{0}) = 0$
Schwefel's problem 1.2 with noise $f_4(\vec{X}) = \sum_{i=1}^{D}\left(\sum_{j=1}^{i} x_j\right)^2 \times \left(1 + 0.4\left\|N(0,1)\right\|\right)$	$-30 \le x_i \le 30$	$f_1(\bar{0}) = 0$
Noisy quartic function $f_5(\vec{X}) = \sum_{i=1}^{D} i x_i^4 + rand[0,1)$	$-30 \le x_i \le 30$	$f_1(\bar{0}) = 0$
Rosenbrock's function	$-30 \le x_i \le 30$	$f_1(\bar{0}) = 0$

ISSN: 2664-9640

$f_6(\vec{X}) = \sum_{i=1}^{D} 100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2$		
Griewank's function $f_7(\vec{X}) = \sum_{i=1}^{D} \frac{x_i^2}{4000} - \prod_{i=1}^{D} \cos(\frac{x_i}{\sqrt{i}}) + 1$	$0 \le x_i \le 600$	$f_1(\bar{0}) = 0$
Ackley's function $f_8(\vec{X}) = -20\exp\left(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2}\right) - \exp\left(\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi x_i)\right) + 20 + e$	$-30 \le x_i \le 30$	$f_1(\bar{0}) = 0$
Rastrigin's function $f_9(\vec{X}) = \sum_{i=1}^{D}\left(x_i^2 - 10\cos(2\pi x_i) + 10\right)$	$-5.12 \le x_i \le 5.12$	$f_1(\bar{0}) = 0$

The test results are shown in Table 2, which includes the average and standard deviation of the optimal solution of each function under different algorithms. After comparison, the better results are shown in bold as follows.

Table 2 The average value and standard deviation of the optimal solution for 300 independent operation

Function		Dimension	Improved adaptation DE	SDE	DE/rand/1
f_1	Mean value	30	1.38E-13	133.9584	1.09E-12
	Standard deviation	30	8.72592E-13	2806.806843	6.3112E-12
f_2	Mean value	30	6.6E-15	2459.44	1.25E-11
	Standard deviation	30	1.08E-15	58197.86	7.48E-11
f_3	Mean value	30	2.23E-15	0.705818	3.06E-15
	Standard deviation	30	3.65E-14	10.68658	1.87E-14
f_4	Mean value	30	0.037425	192551.4	0.134981
	Standard deviation	30	0.474	4973300	0.819638
f_5	Mean value	30	0.0262	1568657	0.301222
	Standard deviation	30	0.1464217	25273214	1.759191
f_6	Mean value	30	2.58E-02	5548.68	38.12193333
	Standard deviation	30	2.59E-02	17923.72376	43.51966
f_7	Mean value	30	6.25084E-12	0.159524	6.29E-13
	Standard deviation	30	1.21447E-12	0.312	1.17E-12

ISSN: 2664-9640

f_8	Mean value	30	1.6844	2.794893	1.6844
	Standard deviation	30	0	3.080693	0
f_9	Mean value	30	6.19E-02	103.5647	83.8317
	Standard deviation	30	6.27E-02	143.5713	84.06448

4. Summary

It is not difficult to find from table 2 that compared with the other two algorithms, the average and standard deviation of the optimal solution of most functions of the improved differential evolution algorithm are smaller than those of other algorithms. In addition to the function f_3 and f_7, de / RAND / 1 itself is easy to fall into local minimum, while the improved algorithm can effectively improve the population diversity and avoid local minimum in numerical experiments.

References

[1]. Qi Feng, Deyun Zhou. Time optimal path planning based on differential evolution algorithm [J]. Computer engineering and application,2005,41(12):74-75.
[2]. Qin A K, Suganthan P N. Self-adaptive differential evolution algorithm for numerical optimization[C]. Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2005, 2-4 September 2005, Edinburgh, UK. IEEE, 2005.
[3]. Koh A. Solving transportation bi-level programs with Differential Evolution[J]. Evolutionary Computation, 2007. CEC2007. IEEE Congress on.IEEE, 2007:2243-2250.
[4]. Intelligent Optimization Algorithm and Its Application [D]. Northeastern University,2012,23-35.
[5]. Xiaojun Bi, Guoan Liu, Jing Xiao. Dynamic adaptive differential evolution algorithm based on new mutation strategy [J]. Computer research and development,2012,49(6):1288-1297.