A Hybrid Flower Pollination Algorithm with Gravity Center Reconstruction and Cauchy Mutation

Jianbin Wang

School of Computer Science and Software, Zhaoqing University, Zhaoqing 526061, China.

5836697@qq.com

Abstract

The flower pollination algorithm (FPA) has some disadvantages, traditional improvement algorithms of FPA are mainly focused on several parameters, an improved FPA algorithm with gravity center reconstruction and Cauchy mutation is proposed. The gravity center reconstruction is used to reduce the optimization space. And then Cauchy mutation is used for random perturbation to increase the diversity of the population, avoid the algorithm falling into local optimum, and improve the ability of global search. The simulation results for several benchmarks show that compared with the traditional search algorithms, the new algorithm has better local convergence speed and solution quality.

Keywords

Flower pollination algorithm, gravity center reconstruction, self-adaptive, Cauchy mutation.

1. Introduction

Flower pollination algorithm (FPA) is an evolutionary algorithm for simulating pollination behavior of flowers [1]. Compared with other evolutionary algorithms, it has simple concept, fewer parameters and is easy to implement. As a new intelligent optimization algorithm, FPA has been successfully applied to the function optimization, meteorological prediction, engineering design, and has good performance on the related fields [2, 3].

At present, many scholars have proposed many different improvement strategies. The improvement methods mainly focus on the parameters and pollination methods in FPA, including probability size, Levy distribution coefficient in global pollination and uniform distribution system in local pollination. Dynamic handover probability is designed to improve the search ability of the algorithm [4]. Based on the sine-cosine algorithm, the local convergence problem is solved by embedding the sine-cosine algorithm into the basic FPA [5]. Wang introduces adaptive step size in cross-pollination to improving search ability [6]. To improve the convergence speed of the algorithm in the later stage, individual Levy flight and gravity is used to update individual position [7]. To a certain extent, these methods above have improved the search ability of the algorithm and achieved good results for specific problems. However, there are still some problems, such as slow evolution in the early stage, difficult convergence of the algorithm, and poor global search ability in the later stage.

A hybrid algorithm of gravity center reconstruction and FPA is presented in this paper. Based on the principle of leverage force balance, the search space is reasonably compressed and simplified by reconstructing the center of gravity of the function, so as to improve the local search ability in the early stage of the algorithm. At the same time, the inert mutation mechanism is introduced in the later stage of the algorithm to enhance the climbing ability of the algorithm and jump out of the local optimum. The simulation experiment of the test function shows the effectiveness of the improved algorithm.

2. Basic FPA

The basic FPA is an evolutionary algorithm inspired by the pollination behavior of flowers in nature. Studies have shown that 90% of the pollination behavior in nature is alienated pollination, while the other 10% is self-pollination. Based on this research, Yang proposed FPA algorithm in 2012[1]. The steps of basic FPA are as follows.

Step1 Setting and initializing the basic parameters of FPA: the number of variables, the range of values of each variable, the conversion probability, the number of populations, the maximum number of iterations, etc.

Step2 Generating initial solution x_i^t : according to fitness function, calculating the fitness value and selecting the current optimal value from the initial solution as the current global optimal solution g^*.

Step3 Producing the random number $rand$, and comparing it with the conversion probability p . If $p > rand$, carrying out the global search according to formula (1), in which L is the step size obeys Levy distribution.

$$x_i^{t+1} = x_i^t + L(x_i^t - g^*) \tag{1}$$

Step4 Otherwise, if there $p < rand$, carrying out the local search according to formula (2), where ε is the step size obeys uniform distribution.

$$x_i^{t+1} = x_i^t + \varepsilon(x_j^t - x_k^t) \tag{2}$$

Step5 Comparing the new solution with the previous one, if the new solution is better than the former one, the former one will be replaced by the new one, otherwise the former one will be retained and the next one will be transferred.

Step6 Determining whether the maximum iteration condition is satisfied, if it is true, stopping the iteration and outputting the optimal solution; otherwise, turning to Step3.

3. Gravity Center Reconstruction

3.1 Relationship between Gravity Center and Optimal Solution

According to the principle of mechanics, the lever principle shows that the center of gravity of an object is always close to the area with dense mass distribution, and the sparser the area is, the farther away it is from the center of gravity. In order to maintain the moment balance, the closer the point with larger mass is from the center of gravity, the farther the point with smaller mass is from the center of gravity. As shown in Figure 1, for a closed area surrounded by a function $f(x)$, C is the maximum value of the function. Through analysis, we know that the "center of gravity" G of $f(x)$ is near the maximum value.

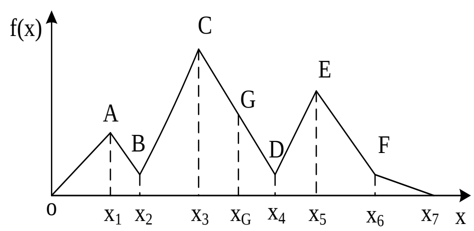

Fig. 1 Relationship between gravity center and optimal solution

3.2 Gravity Center Reconstruction

For complex optimization problems, there are many peaks in the function value, and the probability that the center of gravity of the obtained function falls into the neighborhood of the global optimal value becomes lower, which makes it difficult to find the neighborhood of the global optimal value directly according to the position of the center of gravity of the original function, so it is necessary to reconstruct the center of gravity of the function. In this paper, we use function filling technique to design transformation functions as follows [8].

$$F(x)=\begin{cases} f(x), & f(x)\geq \delta \\ \xi, & f(x)<\delta \end{cases} \tag{3}$$

The result after filling in the transformation function (3) is shown in Figure 2.

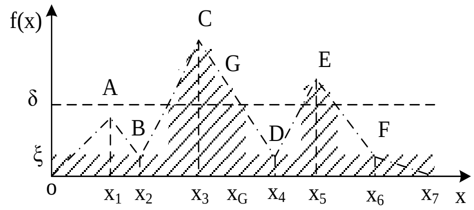

Fig.2 Function filling effect

By adjusting parameter δ, the original optimization space of the function is divided, and the search area larger than δ remains unchanged. The region less than δ is filled with function value ξ to reduce the proportion of local extremum in search space, thus increasing the relative proportion of global optimal extremum, reconstructing the position of center of gravity in function space, and increasing the probability of center of gravity falling in the neighborhood of optimal value.

4. Improvement of FPA

4.1 Improvement of Search Space

Based on the analysis in Section 3, formula (4) is used to iteratively update the position of the center of gravity to realize the reconstruction of the optimization space.

$$x_G(k)=\frac{x_p F(x_p)+\sum_{i=2}^{M(k)} x_i(k)F(x_i(k))}{F(x_p)+\sum_{i=2}^{M(k)} F(x_i(k))} \tag{4}$$

In which, x_p and $F(x_p)$ are the optimal solutions and their function values in $(k-1)^{th}$ iteration, and $M(k)$, population size, is the number of individual solutions generated in the search area in k^{th} iteration.

Assuming that $\eta=O(k)/O(k-1)<1$, then the search range is $O(k)=\eta^k O(0)$ after k iterations, where $O(0)$ is the original search area, we obtain the initial search space of the improved FPA as formula (5).

$$O_{init}=[x_G(k)-\frac{O(k)}{2}, x_G(k)+\frac{O(k)}{2}] \tag{5}$$

Where, $O(k)$ is the search space of k^{th} iteration.

4.2 Cauchy Mutation

Cauchy distribution is a continuous probability distribution with probability density, its function is

$$f(x:x_0,\gamma)=\frac{1}{\pi\gamma[1+(\frac{x-x_0}{\gamma})^2]}=\frac{1}{\pi}[\frac{\gamma}{\gamma^2+(x-x_0)^2}] \tag{6}$$

In which, x_0 is the location parameters of distribution peak, γ is the scale parameters of half width at half maximum. When $x_0=0$, $\gamma=1$, formula (6) can be rewritten as the standard Cauchy distribution in formula (7).

$$f(x)=\frac{1}{\pi}(\frac{1}{x^2+1}) \tag{7}$$

In cross-pollination, the adaptive step size is used instead of the original step size, which makes the algorithm maintain a larger step size in the initial stage and improve the ability of searching the optimal solution. The smaller step size in the later stage is beneficial to improve the accuracy of the algorithm. The new step size is defined as follows.

$$L = L \times \mathrm{e}^{-30\times(T_{\max}/2000)} + 0.0001 \tag{8}$$

In self-pollination, the Cauchy mutation is used for random perturbation, which will help to increase the diversity of the population, avoid the algorithm falling into local optimum, and improve the ability of global search for the best value. The Cauchy mutation is as in formula (9).

$$x_i^{t+1} = \alpha \times Cauchy(0,1) \times x_i^t + \varepsilon(x_j^t - x_k^t) \tag{9}$$

Where, $\alpha = \frac{T_{\max} - t}{T_{\max}}$, $Cauchy(0,1) = \tan[(\varepsilon - 0.5)\pi]$, $\varepsilon \in U[0,1]$.

4.3 Steps of Improved FPA

Based on the above analysis, the steps of the improved FPA are as follows.

Step1 Setting the basic parameters of gravity center reconstruction: population size of gravity center reconstruction $M(k)$, search range compression ratio η, parameters of filling function δ and ξ, range of values of variables, iteration times k.

Step2 Filling the original function space by formula (3) and generating a new center of gravity by formula (4).

Step3 Determining whether the number of iterations is satisfied, if true, turning to the next step; otherwise returning to Step2.

Step4 Initialize the FPA parameters with O_{init} as the initial search space.

Step5 According to the comparison results of $rand$ and conversion probability p, calculating the new solution by formula (1) , formula (2), formula(8) and formula(9).

Step6 Calculating the fitness of the new solution and replacing the corresponding variables if the new fitness is better than the optimal fitness.

Step7 Determining whether the maximum iteration condition is satisfied, if it is true, stopping the iteration and outputting the optimal solution; otherwise, turning to Step4.

5. Simulation and Analysis

To test the solving effect and convergence speed of the improved FPA (IFPA) in this paper, we select four basic single-mode and multi-mode functions in [9] by using particle swarm optimization (PSO), harmony search (HS), FPA and IFPA respectively. The four test functions are as follows.

1) Sphere Function

$$f_1(x) = \sum_{i=1}^{n} x_i^2 ,$$

Where search dimension is 2, search region is [-100，100].

2) Ronsenbrock Function

$$f_2(x) = \sum_{i=1}^{n-1} (100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2) ,$$

Where search dimension is 10, search region is [-30,30].

3) Ackley Function

$$f_3(x) = -20\exp(-0.2\sqrt{(\sum_{i=1}^{n} x_i^2)/n}) - \exp((\sum_{i=1}^{n}\cos(2\pi x_i))/n) + 20 + e,$$

Where search dimension is 10, search region is [-30，30].

4) Griewank Function

$$f_4(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos(\frac{x_i}{\sqrt{i}}) + 1,$$

Where search dimension is 10, search region is [-600, 600].

In order to analyze the performance of each algorithm, each algorithm runs 100 times, the maximum number of iterations is 2000, and the population size is 20. The specific parameters of each algorithm are as follows.

For PSO, $C_1 = 2$, $C_2 = 2$, $W = 0.8$ .For HS, the harmony memory library is $HM = 50$, harmony retention probability is $HMCR = 0.8$, fine-tuning probability is $PAR = 0.5$. For basic FPA, conversion probability is $p = 0.8$, the parameter of standard gamma function is $\lambda = 1.5$. For IFPA, the number of iterations for gravity center reconstruction is $k = 5$, the compression ratio of search space is $\eta = 0.5$, the parameters of filled function are $\delta = 0.01$ and $\xi = 0.001$.

Table 1 shows the results of gravity center reconstruction for Sphere functions. And Tables 2, 3,4 and 5 show the results of different algorithms for each function, where m is the average value of the optimal value obtained by 100 iterations, v is the variance and s are the number of successful searches.

Table 1. Results of gravity center reconstruction for Sphere functions

k	$M(k)$	$x_G(k)$	$O(k)$
1	100	5.67	$-44.33 \le x_1 \le 55.67$
		1.01	$-48.99 \le x_2 \le 51.01$
2	50	-2.43	$-27.43 \le x_1 \le 22.57$
		0.64	$-24.36 \le x_2 \le 25.64$
3	25	0.02	$-11.98 \le x_1 \le 12.02$
		0.12	$-11.88 \le x_2 \le 12.12$
4	12	0.36	$-5.64 \le x_1 \le 6.36$
		-0.04	$-5.04 \le x_2 \le 5.96$
5	6	-0.09	$-3.09 \le x_1 \le 2.91$
		0.01	$-2.99 \le x_2 \le 3.01$

Table 2. Results of Sphere functions

Algorithms	m	v	s
PSO	0.1337	2.3392	65
HS	0.0433	0.9437	72
FPA	0.0474	1.3574	75
IFPA	0	0	94

Table 3. Results of Rosenbrock functions

Algorithms	m	v	s
PSO	7.4637	9.3419	51
HS	4.6308	5.2847	47

FPA	6.9534	3.3574	66
IFPA	1.3274e-010	2.8734e-010	91

Table 4. Results of Arckly functions

Algorithms	m	v	s
PSO	0.8251	49.3652	11
HS	0.0128	0.2039	43
FPA	0.0302	0.7231	36
IFPA	8.7921e-010	9.1672e-010	86

Table 5. Results of Griewank functions

Algorithms	m	v	s
PSO	0.1204	9.3423	8
HS	5.3874e-004	3.7391e-3	23
FPA	0.0956	7.0129	56
IFPA	1.0347e-015	1.4892e-015	80

From Table 1, the center of gravity reconstruction algorithm can compress the search space very well after several iterations. It means that the gravity center reconstruction can reduced the initial search space and greatly improve the initial local search ability of the algorithm.

From Table 2-Table 5, the optimal average value and variance found by IFPA algorithm are better, and the success rate is higher than the other three algorithms. For complex multi-modal functions, the success rate of other algorithms is lower, but IFPA algorithm can still jump out of the local optimal solution more effectively. Therefore, IFPA algorithm is superior to the other three algorithms.

6. Conclusion

In this paper, the basic FPA is improved by Cauchy mutation and a hybrid barycenter reconstructed pollination algorithm (IFPA) is proposed. The gravity center reconstruction method is used to compress the optimization space of simplified function, reduce the search range of the algorithm. And then the Cauchy mutation is used for random perturbation of basic FPA to increase the diversity of the population, avoid the algorithm falling into local optimum, and improve the ability of global search. The simulation results show that the IFPA has good convergence speed and the ability to jump out of local optimum.

Acknowledgements

This work was financially supported by the Natural Science Foundation of Guangdong Province (No. 2016A030313015 and No. 2016A030313846), the Industry University Cooperation Education Project of Ministry of Education of China (No. 201702029053), the Teaching Reform Project of Zhaoqing City (No. ZLGC201842), and the Practical Teaching Reform Project of Zhaoqing University (No. SJJX201812).

References

[1] X.S. Yang: Flower pollination algorithm for global optimization[C]. International Conference on Unconventional Computation and Natural Computation, 2012: 240-249.

[2] A.R. Osama, M. Abdel-Baset: A novel hybrid flower pollination algorithm with chaotic harmony search for solving sudoku puzzles [J]. International Journal of Modern Education & Computer Science, 2014, 6(3): 126-132.

[3] X.S. He, M. Zhang: Applications of multi-objective FPA in electricity portfolio market [J]. Computer Engineering and Applications, 2017, 53 (17):234-240.
[4] R. Salgotra, U. Singh: Application of mutation operators to flower pollination algorithm [J]. Expert Systems with Applications, 2017, 79:112-129.
[5] S. Liu, Q.H. Zhao, S.J. Chen: Flower pollination algorithm based on sine cosine algorithm [J]. Microelectronics and Computer, 2018, 35 (06):84-87.
[6] X.F. Wang, J.Z. Meng: Adaptive flower pollination algorithm based on Cauchy mutation [J]. Intelligent Computer and Applications, 2018,8 (04):106-111.
[7] H.H. Xiao, C.X. Wan, Y.M. Duan and Q.L. TAN: Flower pollination algorithm based on gravity search mechanism [J]. Acta Automatica Sinica, 2017, 43 (04):576-594.
[8] J. Liu, Y.Q. Ye: A new class of filled functions for finding global optimization [J]. Computer Technology and Development, 2010, 20(6):36-38.
[9] S. Rahnamayan, H.R. Tizhoosh, M.M.A. Salama: Opposition-based differential evolution [J]. IEEE Transactions on Evolutionary Computation, 2008, 12(1): 64-79.