A Modified Cuckoo Search Algorithm and Its Applications in Function Optimization

Shao Qiang Ye¹, Fang Ling Wang² and Kai Qing Zhou²

¹College of Mathematics and Statistics, Jishou University, Jishou, 416000, Hunan, China.
²College of Information Science and Engineering, Jishou University, Jishou, 416000, Hunan, China.
Email: kqzhou@jsu.edu.cn

Abstract. A modified Cuckoo search algorithm (MCS) is proposed in this paper to improve the accuracy of the algorithm's convergence by implementing random operators and adapt the adjustment mechanism of the Levy Flight search step length. Comparative experiments reveal that MCS can effectively adjust the search mechanism in the high-dimensional function optimization and converge to the optimal global value.

1. Introduction

Cuckoo Search (CS) algorithm is a new swarm intelligence optimization algorithm proposed by Yang and Deb [1] after analyzing the method of cuckoo's behaviors, such as seeking the other hosts' nests and laying one or many eggs.

In the basic CS, the process of seeking other hosts’ nest is depicted by the Levy flight because the Levy flight is a kind of random walk process consisting of step length small in the short term and occasionally step length long in the long run. On the other hand, the local search accuracy and the convergence efficiency of the CS are both low due to this intermittent jump, and the solution oscillates in the optimal global position in the later iteration, respectively.

Aiming at the inherent problems of the basic CS, a series of improved CS are widely discussed from different viewpoints, such as the step length, parameters, and position (solution) of the bird's nest. Yang et al. [2] proposed the multi-species cuckoo algorithm to enable the co-evolution ability of cuckoo-host in competition. By integrating the annexation and cooperation operators, the diversity of the population, and the optimization ability of the CS algorithm was raised. To solve a constraint problem, Zhang et al. [3] proposed a hybrid CS-DE algorithm to meet the requirement of high precision in constraint problems. The simulation results showed the solution of the presented algorithm has higher precision and more competitive in the process of global optimization. Cheng et al. [4] proposed an adaptive hybrid cuckoo algorithm (AHCS) by employing the mutation operator, evolutionary strategy, and improved Levy flight to solve a CEC2017 optimization problem. The experimental results revealed robustness and the advantages of the AHCS. Abdel-baset et al. [5] proposed a CS-GA, and the simulation experiment proved that the CS-GA has higher accuracy and robustness in solving non-constrained optimization problems. Inspired by the mentioned papers above, a modified CS (MCS) algorithm is proposed in this paper to improve the accuracy of the algorithm's convergence by using an random operators and the Levy Flight search step length adjust approach.
The rest of the paper is organized below. Section 2 recalls the basic CS algorithm. Section 3 discusses the proposed MCS algorithm and the corresponding improved methods. Benchmark problems are used to test the performance of MCS in different dimensions and corresponding experimental results are given in Section 4. Finally, Section 5 summarizes the entire work.

2. Basic CS algorithm
CS algorithm includes some steps, which are a randomly generated initial population, carries out population update, random migration, selection of the best, and judgment of termination conditions and continuously performs the iterative calculation. The basic framework of CS is demonstrated as follows.

Step 1. Initialization of parameter \((N,T,Pa,UB,LB,D)\) and position, define the minimum objective function \(f(x)\).

\[
x^0_i = LB + (UB - LB)\odot \text{rand}(D)
\]  

(1)

Step 2. Calculate the fitness value.

Step 3. Population update by Levy Flight.

\[
x^{t+1}_i = x^t_i + \alpha \odot \text{Levy}(\lambda)
\]  

(2)

Step 4. Random migration.

\[
x^{t+1}_i = \begin{cases} x^t_i + r(x^t_i - x^t_k), & \text{if } r < Pa, \\ x^t_i, & \text{otherwise} \end{cases}
\]  

(3)

Step 5. Selection of the best individual.

Step 6. Check the stopping criterion. Otherwise, return 3.

Where, \(N, T, Pa, UB, LB, D, \alpha, x_i^t\) are the parameters which presents the population size, the maximum iteration numbers, the host bird find cuckoo eggs probabilities, the upper bound, the lower bound, the problem dimension and step factor, the \(i\)-th individual in the \(t\), and \(t\) generation respectively.

Furthermore, \(x_i^t, x_k^t\) denote two random individuals in \(t\) generation \((j \neq k)\).

3. Modified cuckoo search algorithm
In basic CS, global search and local search probability are noted as 3/4 and 1/4[6]. To further improve the efficiency, the implementation process of the random operator dynamic adaptive adjustment is utilizing at first in the MCS. Secondly, the step size was adaptive adjusted in Levy Flight, and the step length will be reduced with the increase of the number of iterations for enhancing the local search ability. The specific improvement measures are listed below.

1. Adaptive adjustment of step length factor.

\[
\alpha = \left(\frac{T-t}{T}\right)\alpha_0
\]  

(4)

\[
x^{t+1}_i = x^t_i + \left(\frac{T-t}{T}\right)\alpha_0 \odot \text{Levy}(\lambda)
\]  

(5)

2. Random operation.

\[
\omega_i = \frac{f(x_i^{t+1}) - f(x_i^t)}{f(x_i^t) - f(x_k^t)}
\]  

(6)

3. Record the execution probabilities of global and local searches according to the discriminant of \(\omega_i\).

\[
L = \frac{\text{total}(L)}{\text{total}(L+G)}
\]  

(7)

\[
G = \frac{\text{total}(G)}{\text{total}(L+G)}
\]  

(8)

Where \(\omega_i\) is related to the fitness values, that is, the search mechanism of CS dynamically changed due to the value of \(\omega_i\) for ensuring the convergence speed of the algorithm. \(\alpha_0, T, L, G\) are parameters to donate the initial step factor \((\alpha_0 = 0.01)\), the current iteration, the local search and the global search probabilities \((L, G \in [0,1]), L+G = 1 \in [0,1]\), respectively.
The relationship among $\omega_l$, $L$, and $G$ is as follows:

First, when $\omega_l > 1$, the local search probability $L$ is enhanced to improve the accuracy of the solution. Second, when $\omega_l < 1$, the global search probability $G$ is enhanced to expand the search space of solution space. Finally, when $\omega_l = 1$, the global search and local search probabilities are kept.

According to the above improved ideas, MCS implementation process is shown in Figure 1.

4. Experimental results and analysis

In this section, to verify the performance of the proposed MCS, CS and MCS are applied to solve different function optimization problem. The related parameters set as follows.

The population number $n=50$, the discovery probability $P_d=0.25$, the corresponding maximum number of iterations ($T$) in different dimensions ($D=10, 30, 50$) are 500, 1000, and 4000, respectively, and all algorithms all are run 30 times independently to obtain the statistical results. Table 1 lists the three selected functions, and Figure 2-4 shows the convergence curves of MCS and CS in solving multimodal functions with different characteristics.

| Function       | Equation                                      | Search Space     | Real Value ($f^*$) |
|----------------|-----------------------------------------------|------------------|--------------------|
| Schwefel       | $f_1 = 418.9829D - \sum_{x_i=1}^{D} \sin(\sqrt{|x_i|})$ | [-500,500]       | 0                  |
| Ackley         | $f_2(x) = -a e^{-\sum_{x_i=1}^{D} \left[ \frac{1}{4}x_i^2 - \cos(\frac{x_i}{\sqrt{2}}) \right] + \sum_{i=1}^{D} \cos(x_i) e^{\frac{2\pi i}{D}}} + a + e$ | [-32,32]         | 0                  |
| Eggholder      | $f_3 = (x_1 + 47) \sin \left( \sqrt{x_1^2 + (x_2 + 47)^2} \right) - x_2 \sin \left( \sqrt{x_1^2 + (x_2 + 47)^2} \right)$ | [-512,512]       | -959.6407          |
Above three pictures indicate that CS and MCS can both converge to the best state in 10D. However, with the increase of dimensions (30D, 50D), MCS shows the advantages of rapid convergence like jumping off a cliff and the convergence accuracy is obviously better than CS.

Through a lot of experimental training, the local and global search probabilities have changed, $L \in [0.25, 0.35], G \in (0.65, 0.75]$. Hence, MCS has a certain adjustment effect on the local search optimization, thus further enhancing the accuracy of the solution.

Table 2 reveals that MCS has a relatively smaller mean value and standard deviation compared with the CS. Meanwhile, MCS obtains the global optimal solution in high-dimensional function optimization and has a certain adaptive ability. The mean and standard deviations ($STDV$) by running 30 times on each function were listed in Table 2.
Table 2. Comparison of experimental results

| Function | CS | MCS |
|----------|----|-----|
|          | D  | MEAN±STDV | MEAN±STDV |
| F1       | 10 | 2.54E+00±6.58E+00 | 4.02E-04±4.07E-04 |
|          | 30 | 2.14E+03±2.17E+03 | 1.84E-02±2.61E-02 |
|          | 50 | 4.84E+03±4.89E+03 | 1.87E+03±1.94E+03 |
| F2       | 10 | 2.51E-01±4.42E-01 | 7.62E-02±7.77E-02 |
|          | 30 | 5.91E+00±6.49E+00 | 5.49E-01±5.54E-01 |
|          | 50 | 1.37E+00±1.37E+00 | 1.12E+00±1.11E+00 |
| F3       | 10 | -9.205659E+02±4.85E+01 | -9.580403E+02±2.22E+00 |
|          | 30 | -9.347136E+02±3.81E+01 | -9.579778E+02±2.17E+00 |
|          | 50 | -9.542290E+02±1.01E+01 | -9.589749E+02±1.25E+00 |

5. Conclusion
In this paper, random operators are used to dynamically change the search strategy in the MCS. Three test functions are utilized to verify the effectiveness of the MCS; the results show that the performance of MCS is much better than that of CS with fast convergence speed and high precision.

References
[1] Yang X S, Deb S. IEEE, 2009: 210-214.
[2] Yang X S, Deb S, Mishra S K. 2018, 10(6): 1085-1095.
[3] Zhang Z, Ding S, Jia W. Engineering Applications of Artificial Intelligence. 2019, 85: 254-268.
[4] Cheng Z, Wang J, Zhang M, et al. IEEE Access, 2019, 7: 145489-145515.
[5] Abdel-Baset M, Hezam I. Applied Mathematics & Information Sciences. 2016, 10(3): 1185-1192.
[6] Yang X S, Deb S. Neural Computing and Applications. 2014, 24(1): 169-174.

Acknowledgement
This work was supported by the National Natural Science Foundation of China (No. 62066016), the Natural Science Foundation of Hunan Province, China (Nos.2020JJ5458 and 2019JJ40234), the Research Foundation of Education Bureau of Hunan Province, China (Nos.18B317 and 19C1535), and the Jishou University Graduate Research and Innovation Project (No. JDY20053).