A Cuckoo Search Algorithm based on Self-adjustment Strategy

To cite this article: Wen Wang and Cong Xie 2018 J. Phys.: Conf. Ser. 1087 022003

View the article online for updates and enhancements.
A Cuckoo Search Algorithm based on Self-adjustment Strategy

Wen WANG, Cong XIE
College of Information Engineering, Guangxi University of Foreign Languages, Nanning, Guangxi, 530222, China
gxwenwang88@sina.cn

Abstract. Cuckoo search algorithm (CS) has been successfully applied to optimization problem. However, CS has the obvious phenomenon of the premature convergence problem and is easily trapped into local optimum. In order to overcome the shortcomings and further improve the performance of CS, a cuckoo search algorithm based on self-adjustment strategy (SACS) is proposed. Eight benchmark test functions are selected to compare with other improved CS algorithm. The simulation results show that the SACS better convergence rate and optimization accuracy than CS, ICS and ASCS when dealing with high-dimensional function optimization problems.

1. Introduction
Many scholars have applied the inspiration gained from the biological community to solve the practical problems, and proposed many meta heuristic swarm intelligent optimization algorithms which were simulated from biological behavior in nature. Ant Colony Algorithm[1], Particle Swarm Optimization[2], Glowworm Swarm Optimization[3], Cuckoo search algorithm [4] are the representatives of such intelligent algorithms. Cuckoo search algorithm derives from hunting and spawning behavior of cuckoo, which was presented by YANG and DEB. The algorithm has been successfully applied to multi-objective optimization[5], face recognition[6], engineering design[7], neural network training[8] and other aspects[9-12].

CS algorithm has many disadvantages, such as slow search speed and low convergence accuracy. Many scholars have proposed improved CS algorithm [8-15]. These improved algorithms improve the performance of CS in some way, but the local search ability of these algorithms is not very strong. In order to overcome the shortcomings and further improve the performance of CS, a cuckoo search algorithm based on self-adjustment strategy (SACS) is proposed in this paper.

2. The Standard Cuckoo Search Algorithm (CS)
In CS algorithm, the following three idealized assumptions are needed to model the way of the cuckoo lays eggs: ① Each cuckoo produces only one egg at a time, and then randomly searches a nest of other birds to hatch it. ② In some random selected nests, the best bird nest will be preserved for the next generation. ③ The number n of nests available is fixed, and the probability of a external bird being found by a master of a bird nest is \( p_r \).

The path seeking and location updating of CS according to Levy flight mode[4] in a multidimensional objective function space, in CS algorithm, as for the \( i \) bird nest, if its location is \( X^i \) at time step \( t \), then at time step \( t+1 \), its new location \( X^{i+1} \) is updated according to the following
formulas:
\[ X_i^{t+1} = X_i^t + a \odot \text{Levy}(\lambda) \]  
\[ \text{Where } a \text{ represents size control of the step, sign } \odot \text{ means point to point multiplications, \text{Levy}(\lambda) is obeys levy distribution, which is } \text{Levy} \sim u = \lambda^{-1}, (1 < \lambda < 3). \]

After location updating, The CS algorithm accords the behavior of the master of bird nest finds an exotic egg and discards it or builds another nest, uses a random rand which obeys uniform distribution compare with \( P_a \), if \( \text{rand} > P_a \) then changes the nest location randomly, finally keep a group better nest location and through several iterations to find the global optimal nest location \( X^* \).

3. The Cuckoo Search Algorithm based on Self-adjustment Strategy(SACS)

3.1. Algorithm Description
The key point of CS is the use of Levy flight strategy. This short distance and the occasional long distance cooperative random search mode make the searching path of the nests of cuckoo is likely to jump from one area to another, so CS is conducive to the global search, the global search ability of CS is very strong. Just because of this mode, CS shows a strong random jump in the search process, so that the searching in the local area which is near each nest of cuckoo is not enough strong, so the convergence speed of CS is slow and the convergence accuracy is not enough high.

In order to overcome the shortcomings and improve the performance of CS, we propose an improved CS algorithm called cuckoo search algorithm based on self-adjustment strategy(SACS), which automatically filters out several poor nest locations at the end of each iteration and automatically produces the same number of new nest locations near the optimum nest location. In this way, the new locations of nests always appear around the current optimal location, which look forward to improving the performance of the algorithm in some ways.

3.2. Mathematical Modeling of SACS
The later period of the SACS in each iteration time, the locations of all nests are sorted, the worst \( m \) nest locations are filtered out, and new \( m \) bird nest locations are created near the current optimum bird nest location. The formula for the new bird nest locations is shown as follows:
\[ X_i^t = X_i^* + \text{randn}(1 - \frac{t}{T_{\text{max}}})^p \]  
\[ \text{Where } X_i^t \text{ is the new location of the nest at time step } t, X_i^* \text{ is the current optimum bird nest location, } \text{randn} \text{ obeys normal distribution with mean value is 0 and variance is 1, } T_{\text{max}} \text{ is the maximum iterations, } p \text{ is regulatory factor, in our simulation, } p = 2. \]

3.3. The Pseudo Code of SACS
Begin
Initialize all parameters: the number \( n \) of bird nests and each nest location \( X(i = 1, 2, ..., n) \), \( P_a \) and other parameters. Set the optimal nest location as the initial value of \( X^* \).

While (\( t \leq \text{the maximum iterations} \))
The nest location is adjusted by formula (1) and get a new set of nest locations \( X_i'(i = 1, ..., n) \).
if( \( f(X_i'(i = 1, ..., n)) > f(X_i^{t+1}(i = 1, ..., n)) \))
then set \( X_i'(i = 1, ..., n) = X_i^{t+1}(i = 1, ..., n) \)
if( \( \text{rand} > P_a \))
A new set of nest locations were obtained by randomly updating. Compared with \( X_i'(i = 1, ..., n) \) in previous step, and better nest location was reserved.
end if
the worst $m$ nest locations are filtered out, and new $m$ bird nest positions are created by formula (2), then find the current optimum bird nest location $X^*$.

end while

End

4. Experimental Simulation and Comparison Analysis

In order to test the performance of SACS, eight benchmark test functions are selected to compared with CS[4], ICS[9] and ASCS[15]. All functions are all complex multimodal function. These algorithms are designed to solve the optimization problem, which try the best to find the minimum of the function. The programming software is matlab2015a, the computer is AMD Athlon (tm) II X2B24 processor and 4GB size memory.

The Parameters setting: the scale of nest is set to 150, Maximum iterations is set to 300, the $P_s$ is set to 0.25 in CS and SACS, the upper limit of $P_s$ is 0.95 and the lower limit of $P_s$ is 0.15, the scaling factor’ maximum is 0.9 and minimum is 0.4 in ICS, the step’ maximum is 1 and minimum is 0.01 in ASCS.

4.1. Test Functions

The first test function:

$$f_1(X) = \sum_{i=1}^{20} [x_i^2 - 10\cos(2\pi x_i) + 10], -5.12 \leq x_i \leq 5.12, \text{ the global minimum value is zero at } x_i = 0.$$  

The second test function:

$$f_2(X) = \frac{1}{4000} \sum_{i=1}^{20} x_i^2 - \prod_{i=1}^{20} \cos \frac{x_i}{\sqrt{i}} + 1, -600 \leq x_i \leq 600, \text{ the global minimum value is zero at } x_i = 0.$$  

The third test function:

$$f_3(X) = 20 + e - 20 \exp(-0.2\sqrt{\frac{1}{50} \sum_{i=1}^{20} x_i^2}) - \exp(\frac{1}{50} \sum_{i=1}^{20} \cos(2\pi x_i)), -30 \leq x_i \leq 30, \text{ the global minimum value is zero at } x_i = 0.$$  

The fourth test function:

$$f_4(X) = \sum_{i=1}^{20} x_i^2 + (\sum_{i=1}^{20} x_i)^2 + (\sum_{i=1}^{20} x_i^4), -5 \leq x_i \leq 5, \text{ the global minimum value is zero at } x_i = 0.$$  

The fifth test function:

$$f_5(X) = \sum_{i=1}^{20} (\sum_{j=1}^{i} x_j)^2, -10 \leq x_i \leq 10, \text{ the global minimum value is zero at } x_i = 0.$$  

The sixth test function:

$$f_6(X) = \sum_{i=1}^{20} x_i^2, -5.12 \leq x_i \leq 5.12, \text{ the global minimum value is zero at } x_i = 0.$$  

4.2. Results and Analysis

In the simulation, for each test function, all algorithms were performed 30 times, the optimal value, the worst value and the average value and the standard deviation are obtained. The results are shown in table 1 and the convergence curves are shown in figure 1 to figure 6.
Table 1 The Results of Simulation

| Function | Algorithm | Optimal Value | Worst Value | Mean | STD |
|----------|-----------|---------------|-------------|------|-----|
|          | CS        | 57.254112     | 81.122164   | 68.583801 | 6.847174 |
| $f_1$    | ICS       | 40.554586     | 69.000787   | 56.596397 | 7.180180 |
|          | ASCS      | 35.111292     | 1.06997e+02 | 65.888736 | 18.389433 |
|          | SACS      | 9.013709      | 37.808880   | 20.203426 | 6.909134 |
|          | CS        | 1.342897      | 2.141451    | 1.816953  | 0.183833 |
|          | ICS       | 1.313191      | 2.049411    | 1.640785  | 0.180084 |
| $f_2$    | ASCS      | 2.179922      | 5.256832    | 3.530251  | 0.664477 |
|          | SACS      | 8.192939e-07  | 0.112826    | 0.033169  | 0.030082 |
|          | CS        | 9.68760991    | 12.417256   | 11.170848 | 0.782026 |
|          | ICS       | 6.36362313    | 9.902224    | 8.100021  | 0.816722 |
| $f_3$    | ASCS      | 8.33995837    | 12.490782   | 10.914286 | 0.950359 |
|          | SACS      | 0.001406      | 1.155179    | 0.079137  | 0.029249 |
|          | CS        | 12.778360     | 37.554731   | 21.788341 | 5.135109 |
|          | ICS       | 16.053228     | 35.690346   | 27.423748 | 6.107941 |
|          | ASCS      | 7.529621      | 22.167215   | 14.374123 | 3.483483 |
|          | SACS      | 0.020556      | 0.233339    | 0.079252  | 0.050268 |
|          | CS        | 12.862348     | 31.299429   | 19.939333 | 4.543290 |
|          | ICS       | 14.435689     | 35.166795   | 24.375231 | 4.957819 |
| $f_4$    | ASCS      | 6.121630      | 16.486340   | 10.758379 | 2.619571 |
|          | SACS      | 0.008604      | 0.071943    | 0.029422  | 0.016150 |
|          | CS        | 1.133977      | 2.773509    | 2.035682  | 0.408107 |
|          | ICS       | 0.814454      | 2.694949    | 1.727106  | 0.408972 |
|          | ASCS      | 0.215483      | 1.650570    | 0.786332  | 0.376080 |
|          | SACS      | 1.839605e-05  | 8.337544e-05| 3.907986e-05| 1.559034e-05|

Figure 1  Convergence Graph of the First Test Function
Figure 2  Convergence Graph of the Second Test Function

Figure 3  Convergence Graph of the Third Test Function

Figure 4  Convergence Graph of the Fourth Test Function
Analysis: It can be seen from Table 1, every evaluation item of SACS is smaller than CS, ICS and ASCS. By analyzing the results listed in Table 1 and the convergence curve simulation diagrams, we can see that SACS converges fast. The convergence speed and optimization accuracy are better than CS, ICS and ASCS. So, the optimization performance of SACS is obviously better than CS, ICS and ASCS.

5. Conclusion and Future Work
Aiming at the problem that the local search of the cuckoo search algorithm is not enough strong, the convergence rate is slow and the convergence accuracy is not enough high in the optimization, a cuckoo search algorithm based on self-adjustment strategy is proposed. The simulation results show that the SACS better convergence rate and optimization accuracy than CS, ICS and ASCS when dealing with high-dimensional function optimization problems. In the future, we may use the ASCS algorithm to optimize the design of the wiring structure to reduce the cost of premises distribution system and other system optimization problems.

Acknowledgment
The work were supported by Guangxi Higher Education Undergraduate Teaching Reform Projects (NO. 2016JGB435), Guangxi Higher Education Undergraduate Teaching Reform Projects (NO.
References

[1] Dorigo M, Bonabeau E, Theraulaz G. Ant algorithms and stigmergy[J]. Future Generation Computer System, 2000, 16(8): 851-871.

[2] Kennedy J, Eberhart R. Particle swarm optimization. Perth: IEEE International Conference on Neural networks, pp. 1995, 1941-1948.

[3] Krishnanand K N, Ghose D. Detection of multiple source locations using a glowworm metaphor with applications to collective robotics. Proceeding of IEEE Swarm Intelligence Symposium, pp.1995, 84-91.

[4] Yang X S, Deb S. Cuckoo search via levy flights[C]. Proceedings of World Congress on Nature & Biologically Inspired Computing India IEEE Publications, USA, 2009, 210-214.

[5] Yang X S, Deb S. Multi-objective cuckoo search for design optimization[J]. Computers and Operations Research, 2013, 40(6): 1616-1624.

[6] Tiwari V. Face recognition based on cuckoo search algorithm[J]. Indian Journal of Computer Science and Engineering, 2012, 3(3): 401-405.

[7] Yang X S, Deb S. Engineering optimization by cuckoo search [J]. International Journal of Mathematical Modeling and Numerical Optimisation, 2010 (4): 330-343.

[8] Valian E, Mohanna S, Tavakoli S. Improved cuckoo search for feed forward neural network training[J]. International Journal of Artificial Intelligence and Application, 2011, 2(3): 36-43.

[9] Hu X X. Improvement cuckoo search algorithm for function optimization problems[J]. Computer engineering and design, 2013, 34(10): 3639-3642.

[10] Ghodrati A, Lotfi S. A hybrid CS/PSO algorithm for global optimization[G]. LNCS 7198:ACIIDS. Part III. Berlin Heidelberg: Springer, 2012: 89-98.

[11] Tuba M, Subotic M, Stanarevic N. Modified cuckoo search algorithm for unconstrained optimization problems[C]. Proceedings of the European Computing Conference, 2011: 263-268.

[12] Layeb A , Boussalia S R. A novel quantum inspired cuckoo search algorithm for bin packing problem[J]. International Journal of Information Technology and computer Science, 2012, 4(5): 58-67.

[13] Valian E, Mohanna S, Tavakoli S. Improved cuckoo search algorithm for global optimization[J]. International Journal of communications and Information Technology, 2011(1): 31-44.

[14] Walton S, Hassan O, Morgan K, et al. Modified cuckoo search: A new gradient free optimization algorithm[J]. Chaos Solitons and Fractals, 2011, 44(9): 710-718.

[15] Zheng H Q, Zhou Y Q. Self-adaptive step cuckoo algorithm[J]. Computer engineering and Applications, 2013, 49(10): 68-71.