

A Hybrid Grasshopper Optimization Algorithm Based on Simulated Annealing

Cong Yang, Huan Li and WenHong Wei⁺

School of Computer, Dongguan University of Technology, Dongguan 523808, China

Abstract. This paper proposed a hybrid grasshopper optimization algorithm to overcome the disadvantages that the grasshopper optimization algorithm was easy to fall into local optimal solution and low accuracy. Firstly, this work used reverse learning strategy to generate the initial population to enhance the global search efficiency and the quality of the solution; secondly, the dynamic compression factor is introduced to replace the linear adaptation of the key parameters in the basic grasshopper optimization algorithm to enhance the global search ability of the algorithm; finally, this paper adapts to the metropolis receiving criterion of simulated annealing algorithm to receive the poor solution with a certain probability, so that the algorithm can be used Enough to jump out of the local optimal solution. Experiments show that the hybrid grasshopper optimization algorithm has stronger global search ability, better accuracy, and can effectively jump out of the local optimal solution.

Keywords: grasshopper optimization algorithm, Hybrid algorithm, compression factor, simulated annealing algorithm.

1. Introduction

The swarm intelligence algorithm is easy to implement and simple to calculate [1]. Moreover, this method only needs the output value of the objective function without its gradient information. A large number of researches on swarm intelligence theory and application methods have proved that swarm intelligence method is a new method that can effectively solve most global optimization problems [2]. In recent years, a large number of swarm intelligence algorithms have been proposed, such as: ant colony optimization [3], particle swarm optimization [4], whale optimization algorithm[5], grey wolf optimizer [6], grasshopper optimization algorithm [7], etc. According to the no free lunch theorem [8], in order to solve different problems, more and more swarm intelligence optimization algorithms are proposed.

Grasshopper optimization algorithm (GOA) [7] is a new swarm intelligence algorithm proposed by Saremi et al. in 2017. It has relatively high search efficiency and speedy convergence velocity, and the special adaptive mechanism of the algorithm itself can well balance the global and local search process. GOA [9] has a strong local development ability, but the global search ability is weak, and it is easy to fall into the local optimum. In response to these problems, Arora [10] et al. proposed a globally optimized chaotic GOA, which replaces the key parameters in the grasshopper optimization algorithm through chaotic mapping, so that the grasshopper optimization algorithm can continue to explore throughout the iteration. Saxena [11] and others proposed an enhanced chaotic GOA to improve the key parameters and improved the chaotic mapping formula to ensure the later development ability.

2. Grasshopper Optimization Algorithm

The grasshopper optimization algorithm is a meta-heuristic bionic optimization algorithm based on the grasshopper population. The model expression of the goad algorithm is as follows:

$$X_i^d = C \left(\sum_{j=1, j \neq i}^N C \frac{ub_d - lb_d}{2} S \left(|x_j^d - x_i^d| \frac{x_j - x_i}{d_{ij}} \right) \right) + \hat{T}_d \quad (1)$$

⁺ Corresponding author. Tel.: 13622672620.
E-mail address: weihw@dgut.edu.cn.

Where X_i^d is the position of the i -th grasshopper on the d -th dimension, and ub_d and lb_d are the upper and lower bounds of the i -th grasshoppers on the d dimension. The decreasing factor C is the comfort zone search parameter, and \hat{T}_d is the target position. S describes the interaction force (attraction and repulsion) of individual grasshoppers. The comfort zone is an area that is neither attractive nor repulsive

$$c = c_{max} - l \frac{c_{max} - c_{min}}{L} \quad (2)$$

$$s(r) = f e^{\frac{-r}{l}} - e^{-r} \quad (3)$$

In the formula, f and l are the attracting strength parameter and the attracting scale parameter. In this paper, $f=0.5$, $l=1.5$

3. Algorithm Description

3.1. Method Description

In order to improve the effectiveness of the algorithm, this work proposed a reverse learning strategy to generate the initial population.

This paper draws on the concept of compression factor proposed by Clerc and Kennedy [12] and adopts the idea of dynamic compression factor. In the iterative process, the size of the compression factor C is controlled by the change of each iteration, as shown in the following formula (5) Show:

$$c = \frac{1}{k} k e^{1 - \frac{1}{L+1-l}} \quad (4)$$

This paper introduced simulated annealing to improve the accuracy of the algorithm and enhance the ability of the algorithm to jump out of the local optimal solution. The annealing function is as follows:

$$T_{K+1} = \begin{cases} \frac{T_K}{\ln(k)} & k \leq m \\ T_K \times 0.99^k & k > m \end{cases} \quad (5)$$

Among them: m is the preset value, this time it is set to 15

The grasshopper position update formula in simulated annealing is as follows:

$$x_{i+1} = \frac{1}{2} \left(x_i - \frac{ub-l}{l^2} + \frac{(ub-l) * 2 * rand}{l^2} \right) \quad (6)$$

3.2. Algorithm Steps

According to the algorithm idea of the grasshopper optimization algorithm, the specific steps of the algorithm in this paper are as follows:

DFOB-GOA algorithm pseudo code

- 1: Set algorithm parameters: population size N , space dimension D , maximum number of iterations L , initial temperature T_0 , end temperature, etc.
- 2: Use the method in 3.1 to generate a new initial population, calculate the fitness value, and find the current best as the target position and target value.
- 3: Use formula (4) to update parameter c .
- 4: Use formula (1) to update the position of the grasshopper so that the distance between any two bodies is in the interval $[1,4]$.
- 5: In the neighborhood where the new solution is obtained, formula (6) is used to generate a new solution, calculate the fitness value, and update the solution according to the Metropolis criterion.
- 6: Determine whether to update the target position and target value according to the fitness value.
- 7: Determine whether the algorithm meets the termination condition, if it is satisfied, it will jump out of the loop and output the target position and target value; otherwise, return to step 3;

4. Experiment Analysis

In order to verify the performance of the algorithm in this paper, the proposed DFOB-GOA and GOA, globally optimized chaotic GOA (divided into CGOA1 and CGOA2 using different chaotic formulas), DA, GAGOA (based on curve adaptive GOA) are compared. The common parameter settings in all optimization algorithms are as follows: the number of populations $N=30$; the number of iterations is 500; the dimension is 30; the original algorithm $c_{min}=0.00001$; $c_{max}=1$;

Each algorithm in the experiment is run independently 50 times, and the performance of the algorithm is evaluated by the average value and variance. This experiment uses 6 benchmark test functions (F1, F5, F6 are Unimodal benchmark functions., F8, F9, F10 are Multimodal benchmark functions) And some CEC2014 test functions.

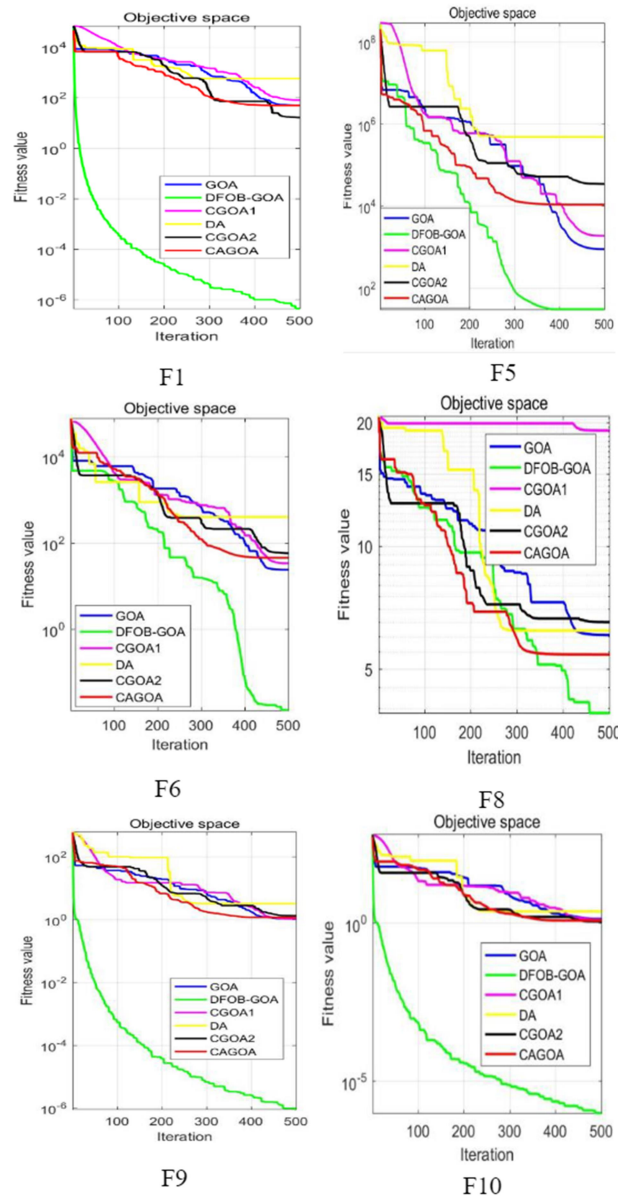


Fig. 1: Benchmark function iteration graph

Table 1,2 shows the comparison results of DFOB-GOA with other algorithms. From the mean and variance in the table, we seen that DFOB-GOA can outperform GOA, CGOA1, CGOA2; and the test function F5~F10 is better than the GOA algorithm, indicating that DFOB -GOA has better convergence and stability, and it can be seen from Figures 1 that the convergence speed of DFOB-GOA is faster than other algorithms.

It can be seen from Table 2 that DFOB-GOA has a similar performance to PSO on the single-peak problem, which is better than GOA, and the performance is better and stable than PSO on the mixed problem.

Table 1: result on CEC2014

algorithm	Target	CEC01	CEC05	CEC17	CEC18	CEC24	CEC25
GOA	Mean	6.19E+07	5.21E+02	3.79E+06	6.08E+03	2.63E+03	2.70E+03
	Std	3.32E+07	6.82E-03	2.03E+06	6.40E+03	3.57E+01	2.63E+00
PSO	Mean	6.69E+06	5.210E+02	2.09E+01	3.97E+03	2.30E+02	2.09E+02
	Std	8.99E+06	6.932E-02	8.52E-02	4.62E+03	6.64E+00	1.64E+00
DFOB-GOA	Mean	3.19E+06	5.20E+02	7.74E+05	6.45E+03	2.60E+03	2.71E+03
	Std	2.06E+06	6.310E-02	2.54E+05	4.80E+03	5.69E+00	9.71E-01

Table 2: result on benchmark test functions

algorithm	Target	F1	F5	F6	F8	F9	F10
GOA	Mean	5.42E+01	1.24E+04	1.96E+01	6.16E+00	1.11E+00	1.30E+00
	Std	35.1675	8796.36	4.54169	0.79135	0.08494	0.16132
DFOB-GOA	Mean	5.18E-07	2.88E+01	1.45E-02	4.99E-04	8.79E-07	8.82E-07
	Std	2.21E-08	0.03015	0.00547	0.00056	1.3E-07	7.2E-08
CGOA1	Mean	2.46E+01	1.61E+04	2.04E+01	1.92E+01	1.12E+00	1.27E+00
	Std	19.35667	25034.6	14.0181	0.30495	0.06104	0.22564
DA	Mean	1734.7933	5.29E+04	1.35E+03	1.25E+01	1.39E+01	8.91E+00
	Std	1252.5545	88158.1	1468.2	6.07348	11.2592	6.49084
CGOA2	Mean	4.22E+01	2.29E+03	5.88E+01	6.52E+00	1.13E+00	1.05E+00
	Std	17.2901	1375.46	55.6954	1.73776	0.0765	0.14491
CAGOA	Mean	2.05E+02	3.34E+03	3.21E+01	6.37E+00	1.27E+00	1.25E+00
	Std	95.71781	2443.33	22.018	1.00502	0.07271	0.1306

5. Conclusion and Future work

This paper proposes a new grasshopper optimization algorithm DFOB-GOA, which uses a reverse learning strategy to increase the diversity and accuracy of the initial population and optimizes the update of individual grasshopper positions during each iteration to ensure that the position is updated. The optimal solution speeds up the convergence speed of the algorithm and increases the probability of finding the optimal solution. The experimental results show that DFOB-GOA has better global search ability and convergence speed compared with the test algorithm.

DFOB-GOA is only able to solve single-objective problems with contentious variable. For future work, binary and multi-objective versions of this algorithm may be developed to solve the discrete and multi-objective problem. Moreover, it is feasible to apply DFOB-GOA to the field of engineering research.

6. Acknowledgements

This work was supported by the Key Project of Science and Technology Innovation 2030 supported by the Ministry of Science and Technology of China (No. 2018AAA0101301), the Key Projects of Artificial Intelligence of High School in Guangdong Province (No. 2019KZDZX1011) and The High School innovation Project (No. 2018KTSCX222).

7. References

- [1] Bansal S, Hota C. Efficient Refinery Scheduling Heuristic in Heterogeneous Computing Systems[J]. Journal of Advances in Information Technology, 2011.
- [2] R. De Castro Edwards Cerqueira de, Salles Evandro Ottoni Teatini, Ciarelli Patrick Marques. A New Approach to Enhanced Swarm Intelligence Applied to Video Target Tracking[J]. Sensors, 2021, 21(5).
- [3] K Hassoune, Dachry W, Moutaoua Kk Il F, et al. Dynamic Parking Guidance Architecture Using Ant Colony Optimization and Multi-agent Systems. 2020.
- [4] Geetha S, Poonthalir G, Vanathi P T. A Hybrid Particle Swarm Optimization with Genetic Operator for Vehicle

Routing Problem[J]. Journal of Advances in Information Technology, 2010, 1(4):181-188.

- [5] Guifang Shen, Shen Guifang. Short-term Traffic Flow Prediction Based on Harmony Search Algorithm Optimized Wavelet Neural Network[J]. Journal of Physics: Conference Series, 2020, 1682(1).
- [6] Mirjalili S, Mirjalili S M, Lewis A. Grey Wolf Optimizer[J]. Advances in Engineering Software, 2014, 69(3):46–61.
- [7] Saremi S, Mirjalili S, Lewis A. Grasshopper Optimisation Algorithm: Theory and application[J]. Advances in Engineering Software, 2017, 105(MAR.):30-47.
- [8] Wolpert D. H., Macready W. G. No free lunch theorems for optimization. IEEE Transactions on Evolutionary Computing, 1997, 1(1): 67-82.
- [9] K. Narsimha Reddy; Polaiiah Bojja. A novel method to solve visual tracking problem: hybrid algorithm of grasshopper optimization algorithm and differential evolution. [J]. Evolutionary Intelligence 2021. PP 1-38.
- [10] Arora S, Anand P. Chaotic grasshopper optimization algorithm for global optimization[J]. Neural Computing & Applications, 2018.
- [11] Saxena A, Shekhawat S, Kumar R. Application and Development of Enhanced Chaotic Grasshopper Optimization Algorithms[J]. Modelling and Simulation in Engineering, 2018, 2018:4945157.1-4945157.14.
- [12] Maurice Clerc and James Kennedy. The particle swarm - explosion, stability, and convergence in a multidimensional complex space. [J]. IEEE Trans. Evolutionary Computation, 2002, 6(1): 58-73.