



# Article An Adaptive Beetle Swarm Optimization Algorithm with Novel Opposition-Based Learning

Qifa Wang, Guanhua Cheng and Peng Shao \*

School of Computer and Information Engineering, Jiangxi Agricultural University, Nanchang 330045, China

\* Correspondence: pshao@whu.edu.cn

Abstract: The Beetle Swarm Optimization (BSO) algorithm is a high-performance swarm intelligent algorithm based on beetle behaviors. However, it suffers from poor search speeds and is prone to local optimization due to the size of the step length. To address this further, a novel improved opposition-based learning mechanism is utilized, and an adaptive beetle swarm optimization algorithm with novel opposition-based learning (NOBBSO) is proposed. In the proposed NOBBSO algorithm, the novel opposition-based learning is designed as follows. Firstly, according to the characteristics of the swarm intelligence algorithms, a new opposite solution is obtained to generate the current optimal solution by iterations in the current population. The novel opposition-based learning strategy is easy to converge quickly. Secondly, an adaptive strategy is used to make NOBBSO parameters self-adaptive, which makes the results tend to converge more easily. Finally, 27 CEC2017 benchmark functions are tested to verify its effectiveness. Comprehensive numerical experiment outcomes demonstrate that the NOBBSO algorithm has obtained faster convergent speed and higher convergent accuracy in comparison with other outstanding competitors.

**Keywords:** optimization; swarm intelligent algorithms; beetle antennae search; beetle swarm optimization; opposition-based learning

# 1. Introduction

Swarm-based intelligent algorithms have been developed to address complicated nonlinear optimization problems more and more in recent years. The reason for this is that there is a variety of optimization problems in engineering which are difficult to be solved effectively in a limited time using traditional optimization methods, and hence it has been one of important research hotspots. Swarm intelligent (SI) algorithms are stochastic search methods which can deal with optimization issues successfully. SI algorithms are designed based on natural occurrences and have been proved to be more excellent in addressing optimization issues than standard algorithms in many cases such as particle swarm algorithm (PSO) [1] developed via mimicking the motion behavior of birds, artificial bee swarm algorithm (ABC) [2] developed based on the forging of bees, ant colony algorithm [3] developed on the basis of the motion of ants and beetle swarm algorithm created based on the movement of beetle swarm.

Beetle swarm optimization (BSO) [4] is a novel swarm intelligent algorithm based on beetle group behavior. Some studies demonstrate that it performs better than previous intelligence algorithms in accuracy and convergence speed. Hence it is a very successful algorithm for addressing optimization problems. Moreover, the algorithm has been widely used in the optimization problems of various disciplines. For example, Wang et al. [5] put forward the improved BSO algorithm based on new trajectory planning and used the algorithm for trajectory planning of robot manipulators. Hariharan et al. [6] mixed PSO and BSO algorithms to propose an adaptive BSO algorithm and applied it to improve BSO algorithm to solve the energy-efficient multi-objective virtual machine integration. Since the search strategy of BSO algorithm is better than PSO algorithm, Mu et al. [7] applied it



Citation: Wang, Q.; Cheng, G.; Shao, P. An Adaptive Beetle Swarm Optimization Algorithm with Novel Opposition-Based Learning. *Electronics* **2022**, *11*, 3905. https:// doi.org/10.3390/electronics11233905

Academic Editor: Claus Pahl

Received: 31 October 2022 Accepted: 23 November 2022 Published: 25 November 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https:// creativecommons.org/licenses/by/ 4.0/). on top of 3D route planning. Singh et al. [8] combined BSO algorithm to propose a heart disease and multi morbidity diagnosis model. Jiang et al. [9] utilized the efficiency of BSO algorithm to localize and quantify structural damage. Zhou et al. [10] put forward an improved BSO algorithm to obtain the shortest path and implementing intelligent navigation control for autonomous navigation robots. Zhang et al. [11] proposed a novel gait multi-objectives optimization strategy based on beetle swarm optimization to solve the lower limb exoskeleton robot.

However, the BSO algorithm still has some drawbacks. When a beetle falls into a local optimal solution, other beetles can gather to the beetle. It tends to make the algorithm skip the global optimal solution, and thus sinks into a local optimal solution. This study proposes a novel improved opposition-based learning method which is more suitable for swarm intelligence algorithms, and it is illustrated in detail in the subsequent Section 3.1 of the study. The original opposition-based learning [12] is an effective intelligent optimization strategy with extensive practical applications, such as the improved particle swarm algorithm [13] that uses opposition-based learning. However, the optimization performance of the opposition-based learning strategy is not so good in the later phase of iterations, so that some variations of the opposition-based learning algorithm have emerged such as the refraction opposition-based learning model based on refraction principle [14] and the elite opposition-based learning [15–17].

Therefore, in this study, a novel opposition-based learning method is proposed to enhance the performance of the original opposition-based learning. In addition, an adaptive strategy is proposed to achieve a better balance on the selection between the BSO algorithm and particle swarm algorithm to enhance its optimization. In the future, the proposed algorithm can be used to solve engineering problems to verify its performance, such as a robust fuzzy control approach for path-following control of autonomous vehicles [18].

The remainder of the study is constructed as below. Firstly, Section 2 demonstrates the related work of this study including the beetle swarm algorithm and the opposition-based learning. After that, the design of the NOBBSO algorithm is presented in Section 3, which includes the utilization of the opposition-based learning in the algorithm and an adaptive strategy is proposed. Section 4 analyzes the experiments, including parameter settings and test functions and convergence analysis. Section 5 discusses the merits and demerits of the NOBBSO algorithm. Finally, a summary is presented in Section 6.

#### 2. Related Preparatory Knowledge

## 2.1. Beetle Swarm Optimization

## 2.1.1. Thought of Beetle Antennae Search

Beetle antennae search (BAS) [19] algorithm is a beetle behavior-based algorithm with the following basic idea: the two antennae of the beetle, like those of most insects, are its primary chemical receptors. Its antennae play a key role in helping it discover food by receiving up signals from its partners. When a signal is received by the antennae of the beetle, it compares the signal intensity between the two antennae and moves towards to the direction of the stronger signal.

Beetle and tentacle are regarded as particles based on their behaviors, and a mathematical model is developed for them. The random search direction of the beetle is formulated as the following Equation (1)

$$\vec{b} = \frac{rands(n,1)}{\|rands(n,1)\|}$$
(1)

where *b* is the beetle search direction, and  $rands(\cdot)$  denotes a random function. *n* denotes a point in that dimension.

The following is the relationship between the antenna and the beetle, which is represented by Equations (2) and (3).

$$x_r^t = x^t + \frac{d^t}{2} \cdot \overrightarrow{b}$$
 (2)

$$x_l^t = x^t - \frac{d^t}{2} \cdot \overrightarrow{b}$$
(3)

where  $x^t$  denotes the position of the *t*-th generation, and  $d^t$  represents the distance between the beetle and the antenna.  $x_r^t$  and  $x_l^t$  represent the positions of the right and left tentacles of beetles, respectively.

The movement of the beetle is then abstracted into the following Equation (4).

$$x^{t+1} = x^t + \delta^t \cdot \overrightarrow{b} \cdot sign(f(x_r^t) - f(x_l^t))$$
(4)

where  $\delta^t$  denotes the step size of the search of the beetle, and its value decreases as the numbers of searching increase. The term  $x^t$  denotes the position of the beetle at the *t*-th generation. The term  $x^{t+1}$  represents the position of the beetle at the next generation. The function  $sign(\cdot)$  denotes the symbolic function and the function  $f(\cdot)$  is used to calculate the signal condition at that instant.

The size of  $d^t$  is fluctuated with the one of  $\delta^t$  in this process, which is formulated as the following Equations (5) and (6).

$$\delta^{t+1} = 0.95 \cdot \delta^t \tag{5}$$

$$d^t = \frac{\delta^t}{c_2} \tag{6}$$

where  $c_2$  is a constant.

## 2.1.2. Beetle Swarm Optimization Principle

Beetle swarm optimization is an improved beetle antennae algorithm with better optimization performance. Because the BAS algorithm is not particularly expert in dealing with multi-dimensional functions, the creator of the BSO algorithm employs the idea of the swarm intelligent algorithm to improve its performance.

The general thought of the beetle swarm optimization algorithm is described as below. Beetles denote candidate solutions for a given optimization problem, and they can exchange their fitness information with each other in the same way as a particle swarm optimization algorithm does in the population. However, the distances and directions of beetles are governed by their speed and antenna information.

In mathematical terms, similar to the PSO algorithm, there are *n* beetles  $X = (X_1, X_2, \dots, X_n)$  in the searching space in the *S* dimension. The position information of the *i*-th beetle is expressed as  $X_i = (X_{i1}, X_{i2}, \dots, X_{is})$ , and the velocity of the *i*-th beetle is expressed as  $V_i = (V_{i1}, V_{i2}, \dots, V_{is})$ , and the optimal *i*-th beetle is expressed as  $P_i = (P_{i1}, P_{i2}, \dots, P_{is})$ . The optimal beetle in a group of beetles is expressed as  $P_g = (P_{g1}, P_{g2}, \dots, P_{gs})$ . The movement of beetles is described as the following Equation (7).

$$X_{is}^{t+1} = X_{is}^{t} + \lambda V_{is}^{t} + (1 - \lambda)\xi_{is}^{t}$$
(7)

where the parameter  $s = 1, 2, \dots, S$ ,  $i = 1, 2, \dots, n$ , and t is the number of the current iteration, and  $V_{is}^t$  represents the speed of the beetle, and  $\xi_{is}^t$  represents the increment of the movement of the beetle. The parameter  $\lambda$  is a constant.

The velocity formula can be written as the following Equation (8).

$$V_{is}^{t+1} = wV_{is}^t + c_1r_1(P_{is}^t - X_{is}^t) + c_2r_2(P_{gs}^t - X_{gs}^t)$$
(8)

where  $c_1$  and  $c_2$  are positive numbers.  $r_1$  and  $r_2$  are random numbers ranging from 0 to 1. w is a weight value which is an adaptive number with the following Equation (9).

$$w = w_{\max} - \frac{w_{\max} - w_{\min}}{T} \cdot t \tag{9}$$

where  $w_{\text{max}}$  and  $w_{\text{min}}$  represent the maximum and minimum values respectively. The term t is the number of the current iteration, and T the number of the maximum iteration.

The above parameter  $\xi$  is defined as the following Equation (10).

$$\xi_{is}^{t+1} = \delta^t \cdot V_{is}^t \cdot sign(f(X_{rs}^t) - f(X_{ls}^t))$$
(10)

where the parameter  $\delta$  denotes the size of step, and the terms  $X_{rs}^t$  and  $X_{ls}^t$  mean the positions of the antennas, respectively.

#### 2.2. Opposition-Based Learning

The following is the general premise of opposition-based learning, which is a strong algorithm optimization strategy.

When looking for optimal solutions, we will start with a current optimal solution x and attempt to get it as close to the real optimal solution as possible. However, sometimes the real optimal solution may be far beyond the current optimal solution, which will consume a lot of time to seek for the optimal solution. Hence we can employ the current optimal solution in the current population for performing the opposition-based learning to generate a new solution  $x_{old}$  which can be formulated as Equation (11).

$$x_{old} = a + b - x \tag{11}$$

where the terms *a* and *b* are the upper and lower bounds of *x*, respectively.

When  $x_{old}$  is greater than x, it is designated as the current optimal solution, which works. However, the opposition-based learning has significant drawbacks, which can only improve the algorithm in the early step while the algorithm is prone to falling into the local extreme solution in the later step.

## 3. The Proposed Algorithm Design

# 3.1. Novel Opposition-Based Learning

The opposition-based learning can produce a better solution for the searched solution, but these better solutions may be dispersed by the opposition-based learning at this point, which prevents the algorithm from finding the optimal solution in later iterations because most of beetles would gather at a point.

The opposition-based learning can be described as follows. New opposite solutions are obtained from current solutions by the way of symmetry at the middle point when the opposition-based learning formula is embodied in coordinates as Figure 1. In a search area, opposition-based learning is employed to make a new solution.



Figure 1. The process of the novel opposition-based learning.

Next the novel opposition-based learning is put forward. Beetle individuals have the best location information (the optimal individual) in the BSO algorithm. Therefore, in the novel opposition-based learning process, the new solution is generated by a method similar to opposition-based learning. However, the difference between them is that the new solution in the novel opposition-based learning comes from the best location information (the optimal individual), not the middle point. In this method, when the current solution is far away from the optimal area, the new solution generated by this strategy can have a greater probability to be closer to the optimal solution to accelerate the convergence speed. Furthermore, because most beetles have already been near to the present optimal position in the later stages of iteration, the new solution obtained by continuing performing the novel opposition-based learning will not be far from the current optimal location.

The enhanced opposition-based learning formula can be defined as the following Equation (12) using the BSO algorithm.

$$x_{new} = 2P_g - x \tag{12}$$

where the term  $P_g$  denotes the optimal solution of all current individuals, *x* denotes that the optimal solution of the current individual and  $x_{new}$  denotes a newly generated optimal solution.

When the beetle swarm algorithm calculates a solution every time, the Equation (12) is employed to calculate a new solution to enhance population diversity.

## 3.2. The Adaptive Strategy

The displacement in the original BSO algorithm is weighted by the parameter  $\lambda$ , as can be seen from displacement formula (seen from Equation (7)). However, there is a key point here in that the value of  $\lambda$  is always a constant value, which results in a problem that beetles do not show greater group behavior that generated by randomly distribution in various positions in the searching space in the early stages of iteration. The group behavior allows them to swiftly gather to the optimal beetle, and most beetles will congregate near the optimal solution. At this moment, beetles should exhibit more individual behavior, so that instead of constantly traveling toward the current optimal beetle, they move according to individual behavior, which allows them to swiftly locate the optimal solution.

Therefore, based on the above idea, the formula for calculating  $\lambda$  can be defined as the following Equation (13):

$$\lambda = \lambda_{\max} - \frac{(\lambda_{\max} - \lambda_{\min})}{T} \cdot t \tag{13}$$

where t is the number of the current iteration, and T is the number of the maximum iteration.

#### 3.3. Description of the Designed Algorithm

The performing process of the BSO algorithm is described as the following Algorithm 1:

## Algorithm 1: BSO

```
S1: Initialize the beetle X<sub>i</sub>, and population velocity v;
```

- S2: Set step size, speed boundary, population size and maximum iteration and so on;
- S3: The fitness of beetles is calculated;
- S4: While (*t* <= *T*)
- S5: Equation (9) is used to calculate the weight *w*;
- S6: Update d using the Equation (6);
- S7: For every single beetle

S9: Equation (10) is used to calculate the increment of the movement

S11: Use the Equation (7) to update the beetle position;

- S13: The fitness of each beetle is computed;
- S14: Record and store the current location of the beetle;
- S15 t = t + 1;
- S16: End while

S8: Equations (2) and (3) are used to obtain the positions of left and right antennas of beetles, respectively;

S10: Equation (8) is used to update beetle velocity *V*;

S12: End for

The performing process of the NOBBSO algorithm is described as the following Algorithm 2:

- S2: Set step size, speed boundary, population size and maximum iteration and so on;
- S3: The fitness of beetles is calculated;

S4:While  $(t \le T)$ 

- S5: Equation (9) is used to calculate the weight *w*;
- S6: Update  $\lambda$  using the Equation (13);
- S7: Update *d* using the Equation (6);
- S8: For every single beetle

S9: Equations (2) and (3) are used to obtain the positions of left and right antennas of beetles, respectively;

- S10: Equation (10) is used to calculate the increment of the movement
- S11: Equation (8) is used to update beetle velocity *V*;
- S12: Use the Equation (7) to update the beetle position;
- S13: Calculate the position of a new beetle using the Equation (12);
- S14: If the fitness of the new position is less than that of the beetle position
- S15: Update beetle position with a new beetle position;
- S16: End if
- S17: End for
- S18: The fitness of each beetle is computed;
- S19: Record and store the current location of the beetle;
- S20: t = t + 1;
- S21: End while

The performing process of the BSO and the NOBBSO algorithm is described as the following Figure 2:



**Figure 2.** Flowchart of BSO and NOBBSO (the left is the BSO flow chart and the right is the NOBBSO flow chart).

# 4. The Experimental Verification

# 4.1. Related Parameter Settings and Test Functions

27 CEC2017 [20] benchmark functions (exclude f17, f20, f29) are used to test and compare the improved BSO algorithm with the original BSO algorithm to validate the performance of the improved BSO algorithm. The reason that these three functions are missing is that the original CEC2017 does not have a Python version, and here a third-party modified Python version of the CEC2017 is used (not include these three functions). Other two BSO algorithm are employed as competitors where BSO is the original beetle swarm algorithm and LBSO is the beetle swarm algorithm improved by Lévy flight. The initial population size of beetles is 50, and the dimension size of benchmark functions are set as 30. The number of iterations is 1000, and  $\lambda_{max}$  and  $\lambda_{min}$  are 1 and 0.4 respectively, and each function is repeated to run for 30 times. Related information of test problems is introduced briefly in Table 1.

Types	No.	Functions	Optimal
TIP	f1	Shifted and Rotated Bent Cigar	100
UF	f2	Shifted and Rotated Sum of Different Power	200
	f3	Shifted and Rotated Zakharov	300
	f4	Shifted and Rotated Rosenbrock	400
	f5	Shifted and Rotated Rastrigin	500
CME	f6	Shifted and Rotated Expanded Scaffer's F6	600
SMF	f7	Shifted and Rotated Lunacek Bi_Rastrigin	700
	f8	Shifted and Rotated Non-Continuous Rastrigin	800
	f9	Shifted and Rotated Lévy	900
	f10	Shifted and Rotated Schwefel	1000
	f11	Hybrid Function 1 (N = 3)	1100
	f12	Hybrid Function 2 ( $N = 3$ )	1200
	f13	Hybrid Function 3 ( $N = 3$ )	1300
LIE	f14	Hybrid Function 4 ( $N = 4$ )	1400
ПГ	f15	Hybrid Function 5 ( $N = 4$ )	1500
	f16	Hybrid Function 6 ( $N = 4$ )	1600
	f18	Hybrid Function 6 ( $N = 5$ )	1800
	f19	Hybrid Function 6 (N = $5$ )	1900
	f21	Composition Function 1 ( $N = 3$ )	2100
	f22	Composition Function 2 ( $N = 3$ )	2200
	f23	Composition Function 3 $(N = 4)$	2300
	f24	Composition Function 4 $(N = 4)$	2400
CF	f25	Composition Function 5 ( $N = 5$ )	2500
	f26	Composition Function 6 ( $N = 5$ )	2600
	f27	Composition Function 7 ( $N = 6$ )	2700
	f28	Composition Function 8 ( $N = 6$ )	2800
	f30	Composition Function 10 (N = 3)	3000
		Search Range: [-100, 100] <sup>S</sup>	

 Table 1. Test functions.

It is worth noting that UF denotes the unimodal functions and SMF denotes the simple multimodal functions, and HF denotes the hybrid functions and CF denotes the composition functions.

#### 4.2. The Experimental Method

# 4.2.1. Analysis of NOBBSO with Different Parameters

To test the performance of the algorithm in relation to the parameters, the following experiments investigate the effect of different parameters on the performance of the algo-

rithm. In addition, the BSO and NOBBSO algorithms are compared on CEC2017 function 1 (Shifted and Rotated Bent Cigar). Population (Pop) sizes of 10, 20 and 30 are tested. Dimension (Dim) sizes of 10, 20 and 30 also are tested. The number of iterations is 1000.

The minimum values from Table 2 show that NOBBSO is easier to find the global optimal solution than the original BSO. However, it also shows some problems about NOBBSO. As the novel opposition-based learning generates a new solution, the solution is uncertain. As a result, the evaluation indicator (Std) of NOBBSO is worse than BSO.

Parameters	Algorithms	Min	Max	Avg	Std
Pop = 10	BSO	$2.63  imes 10^7$	$1.89  imes 10^{12}$	$6.32  imes 10^9$	$4.51  imes 10^{10}$
Dim = 10	NOBBSO	$1.02  imes 10^5$	$1.89  imes 10^{12}$	$7.47  imes 10^9$	$4.81 imes10^{10}$
Pop = 10	BSO	$6.08  imes 10^9$	$2.85  imes 10^{12}$	$7.76 imes10^{10}$	$8.93\times10^{10}$
Dim = 20	NOBBSO	$1.66 imes10^7$	$2.45 imes10^{12}$	$3.14 imes10^{10}$	$8.86 imes10^{10}$
Pop = 10	BSO	$1.45  imes 10^{11}$	$4.29  imes 10^{12}$	$2.78 imes10^{11}$	$1.59  imes 10^{11}$
Dim = 30	NOBBSO	$4.00 imes10^9$	$4.17 imes10^{12}$	$1.41  imes 10^{11}$	$1.72  imes 10^{11}$
Pop = 20	BSO	$2.67  imes 10^4$	$2.00  imes 10^{12}$	$3.07  imes 10^9$	$4.82  imes 10^{10}$
Dim = 10	NOBBSO	$8.33  imes 10^4$	$1.80 imes10^{12}$	$3.20 \times 10^9$	$4.81 imes10^{10}$
Pop = 20	BSO	$1.32 \times 10^9$	$2.89  imes 10^{12}$	$2.47 imes10^{10}$	$8.23\times10^{10}$
Dim = 20	NOBBSO	$1.76 imes10^6$	$2.79 imes10^{12}$	$1.20 imes10^{10}$	$8.42  imes 10^{10}$
Pop = 20	BSO	$2.65  imes 10^{10}$	$5.16  imes 10^{12}$	$9.20 imes10^{10}$	$1.51  imes 10^{11}$
Dim = 30	NOBBSO	$3.42  imes 10^7$	$4.30 imes10^{12}$	$3.47 imes10^{10}$	$1.50 imes10^{11}$
Pop = 30	BSO	$1.16  imes 10^2$	$2.20  imes 10^{12}$	$2.69  imes 10^9$	$5.12  imes 10^{10}$
Dim = 10	NOBBSO	$2.02  imes 10^4$	$1.93 imes10^{12}$	$2.47 imes10^9$	$4.87 imes10^{10}$
Pop = 30	BSO	$3.44  imes 10^8$	$2.66  imes 10^{12}$	$1.26  imes 10^{10}$	$8.15 imes10^{10}$
Dim = 20	NOBBSO	$1.02 imes10^{6}$	$2.94  imes 10^{12}$	$9.68 imes10^9$	$8.31\times10^{10}$
Pop = 30	BSO	$8.18 imes10^9$	$4.71  imes 10^{12}$	$5.36 imes10^{10}$	$1.45  imes 10^{11}$
Dim = 30	NOBBSO	$7.86  imes 10^6$	$4.82  imes 10^{12}$	$2.65  imes 10^{10}$	$1.50  imes 10^{11}$
Statistica	al results	7/9	6/9	7/9	4/9

**Table 2.** The analysis of parameters.

#### 4.2.2. Impact of Adaptive Strategies on Algorithms

In order to figure out the impact of adaptive strategy on NOBBSO, population and dimension parameters of both algorithms in this experiment are 30 while the parameter  $\lambda$  is set to different values as the following Table 3.

Table 3. The analysis of adaptive strategy.

λ	Min	Max	Avg	Std
0.1	$6.56 imes10^{10}$	$5.27  imes 10^{12}$	$1.24\times10^{11}$	$1.38\times10^{11}$
0.25	$2.41  imes 10^{10}$	$4.76\times10^{12}$	$7.26  imes 10^{10}$	$1.40  imes 10^{11}$
0.5	$1.00  imes 10^7$	$4.46\times 10^{12}$	$1.76  imes 10^{10}$	$1.45  imes 10^{11}$
0.75	$5.20  imes 10^6$	$4.80  imes 10^{12}$	$1.77\times10^{10}$	$1.51\times10^{11}$
0.9	$4.46 imes10^6$	$4.45\times10^{12}$	$1.89  imes 10^{10}$	$1.50 imes10^{11}$
Adaptive (Equation (13))	$3.73 imes10^6$	$5.00  imes 10^{12}$	$1.91  imes 10^{10}$	$1.49  imes 10^{11}$

As can be seen from Table 3, the optimal solution with the adaptive strategy is superior to the algorithm without this strategy. Since  $\lambda$  is a constant value, the solution obtained with

the increase of the parameter is more accurate. Therefore, the parameter is set adaptively to a changing value from the larger value to the smaller value, which makes the parameter the maximum value at the beginning of the iteration, and then decreases sequentially. Experiments show that adaptive strategy is very effective in the application of BSO.

# 4.2.3. Accuracy Results of Different Comparison Algorithms

Next, four evaluation indexes including Max, Min, Avg and Std which are the maximum value, minimum value, mean value and standard deviation respectively, and the test results are analyzed according to them, which are listed in Table 4.

Functions	Algorithms	Min	Max	Avg	Std
	BSO	$2.83 \times 10^9$	$4.42  imes 10^{12}$	$2.80  imes 10^{10}$	$1.44  imes 10^{11}$
f1	LBSO	$1.89  imes 10^7$	$4.65  imes 10^{12}$	$2.20  imes 10^{10}$	$1.62\times10^{11}$
	NOBBSO	$1.62  imes 10^{6}$	$4.60  imes 10^{12}$	$1.90 imes10^{10}$	$1.45\times10^{11}$
	BSO	$7.73  imes 10^{20}$	$1.89 imes10^{73}$	$1.61  imes 10^{69}$	$1.28  imes 10^{71}$
f2	LBSO	$8.43  imes 10^{13}$	$3.41  imes 10^{73}$	$3.53 imes10^{69}$	$2.77 imes10^{71}$
	NOBBSO	$3.32  imes 10^7$	$7.87  imes 10^{73}$	$7.32  imes 10^{69}$	$6.03 imes10^{71}$
	BSO	$1.81  imes 10^4$	$2.49\times 10^{16}$	$9.55  imes 10^{12}$	$3.58\times10^{14}$
f3	LBSO	$1.19 \times 10^3$	$2.56 imes10^{16}$	$1.11  imes 10^{13}$	$4.13\times10^{14}$
	NOBBSO	$3.11  imes 10^2$	$1.94 imes10^{16}$	$8.71 imes10^{12}$	$3.17 imes10^{14}$
	BSO	$5.47 \times 10^2$	$5.38  imes 10^5$	$1.41 \times 10^3$	$1.36  imes 10^4$
f4	LBSO	$4.69 \times 10^2$	$6.41 \times 10^5$	$1.46 \times 10^3$	$1.42  imes 10^4$
	NOBBSO	$4.59 imes10^2$	$5.60 \times 10^5$	$1.22  imes 10^3$	$1.29  imes 10^4$
	BSO	$6.45  imes 10^2$	$2.28  imes 10^3$	$7.08 \times 10^2$	$6.42  imes 10^1$
f5	LBSO	$6.36  imes 10^2$	$2.27  imes 10^3$	$7.02 \times 10^2$	$6.98  imes 10^1$
	NOBBSO	$6.13 imes10^2$	$2.20  imes 10^3$	$6.86 imes10^2$	$6.96  imes 10^1$
	BSO	$6.47  imes 10^2$	$1.03 \times 10^3$	$6.68  imes 10^2$	$1.61  imes 10^1$
f6	LBSO	$6.46 \times 10^2$	$1.02 \times 10^3$	$6.72 \times 10^2$	$1.59  imes 10^1$
	NOBBSO	$6.39 imes10^2$	$1.01  imes 10^3$	$6.60 imes10^2$	$1.70  imes 10^1$
	BSO	$1.01 \times 10^3$	$7.59 \times 10^3$	$1.21  imes 10^3$	$2.36  imes 10^2$
f7	LBSO	$9.26  imes 10^2$	$7.34 \times 10^3$	$1.05  imes 10^3$	$2.47  imes 10^2$
	NOBBSO	$9.10 imes10^2$	$8.07  imes 10^3$	$1.08 \times 10^3$	$2.29  imes 10^2$
	BSO	$8.80  imes 10^2$	$2.40 imes10^3$	$9.50  imes 10^2$	$6.44  imes 10^1$
f8	LBSO	$8.76  imes 10^2$	$2.18 imes10^3$	$9.51  imes 10^2$	$6.34 imes10^1$
	NOBBSO	$8.83  imes 10^2$	$2.29 \times 10^3$	$9.45 imes10^2$	$5.95  imes 10^1$
f9 -	BSO	$2.83  imes 10^3$	$1.89  imes 10^5$	$5.08  imes 10^3$	$5.02  imes 10^3$
	LBSO	$2.77  imes 10^3$	$1.96  imes 10^5$	$5.12  imes 10^3$	$5.30  imes 10^3$
	NOBBSO	$2.57  imes 10^3$	$1.99  imes 10^5$	$5.00 imes10^3$	$5.14 imes10^3$
	BSO	$4.70 \times 10^3$	$1.71 \times 10^4$	$6.93  imes 10^3$	$1.33  imes 10^3$
f10	LBSO	$4.00  imes 10^3$	$1.67  imes 10^4$	$5.81  imes 10^3$	$1.13  imes 10^3$
	NOBBSO	$4.29  imes 10^3$	$1.69  imes 10^4$	$6.23  imes 10^3$	$1.15  imes 10^3$

Table 4. Accurate results of three algorithms when D is 30.

 Table 4. Cont.

Functions	Algorithms	Min	Max	Avg	Std
	BSO	$1.31  imes 10^3$	$6.65  imes 10^{11}$	$3.11  imes 10^8$	$1.09  imes 10^{10}$
f11	LBSO	$1.16  imes 10^3$	$6.61  imes 10^{11}$	$3.29  imes 10^8$	$1.14 imes10^{10}$
	NOBBSO	$1.19  imes 10^3$	$9.15  imes 10^{11}$	$4.27  imes 10^8$	$1.52  imes 10^{10}$
	BSO	$2.96  imes 10^7$	$1.34  imes 10^{12}$	$2.89  imes 10^9$	$3.67  imes 10^{10}$
f12	LBSO	$1.67  imes 10^6$	$1.34  imes 10^{12}$	$4.00  imes 10^9$	$4.09  imes 10^{10}$
	NOBBSO	$1.51 imes10^{6}$	$1.43  imes 10^{12}$	$2.14 imes10^9$	$3.70  imes 10^{10}$
	BSO	$9.80  imes 10^3$	$2.66  imes 10^{12}$	$3.36  imes 10^9$	$7.22  imes 10^{10}$
f13	LBSO	$1.37  imes 10^4$	$2.86  imes 10^{12}$	$4.97 imes10^9$	$7.27 imes10^{10}$
	NOBBSO	$7.43  imes 10^4$	$2.68  imes 10^{12}$	$3.04 imes10^9$	$6.97 imes10^{10}$
	BSO	$1.76  imes 10^3$	$3.72  imes 10^{10}$	$2.48 imes10^7$	$8.00  imes 10^8$
f14	LBSO	$1.67  imes 10^3$	$3.75  imes 10^{10}$	$2.54 imes10^7$	$8.11  imes 10^8$
	NOBBSO	$1.96  imes 10^3$	$3.89  imes 10^{10}$	$2.37 imes10^7$	$7.71 imes10^8$
	BSO	$5.74  imes 10^3$	$1.56  imes 10^{12}$	$1.21  imes 10^9$	$3.40  imes 10^{10}$
f15	LBSO	$2.00  imes 10^3$	$1.50  imes 10^{12}$	$1.54 imes10^9$	$3.59 imes10^{10}$
	NOBBSO	$1.04  imes 10^4$	$1.54  imes 10^{12}$	$1.29  imes 10^9$	$3.54 imes10^{10}$
	BSO	$2.68  imes 10^3$	$5.39  imes 10^5$	$3.60  imes 10^3$	$9.43  imes 10^3$
f16	LBSO	$2.16  imes 10^3$	$4.69  imes 10^5$	$3.42  imes 10^3$	$1.00  imes 10^4$
	NOBBSO	$2.09  imes 10^3$	$5.45  imes 10^5$	$3.22 \times 10^3$	$1.02  imes 10^4$
	BSO	$4.30 imes10^4$	$9.96 imes10^{10}$	$7.47  imes 10^7$	$2.32 \times 10^9$
f18	LBSO	$1.56  imes 10^4$	$1.12  imes 10^{11}$	$8.17  imes 10^7$	$2.50  imes 10^9$
	NOBBSO	$3.66  imes 10^4$	$1.22  imes 10^{11}$	$8.35  imes 10^7$	$2.61  imes 10^9$
	BSO	$8.49\times 10^3$	$1.85  imes 10^{12}$	$1.40  imes 10^9$	$3.81  imes 10^{10}$
f19	LBSO	$2.11  imes 10^3$	$1.63  imes 10^{12}$	$1.57  imes 10^9$	$3.71  imes 10^{10}$
	NOBBSO	$7.15  imes 10^3$	$1.64  imes 10^{12}$	$1.26 imes10^9$	$3.54 imes10^{10}$
	BSO	$2.41  imes 10^3$	$1.60 \times 10^4$	$2.49  imes 10^3$	$1.97 \times 10^2$
f21	LBSO	$2.44  imes 10^3$	$1.37  imes 10^4$	$2.53  imes 10^3$	$1.90  imes 10^2$
	NOBBSO	$2.38 imes10^3$	$1.11  imes 10^4$	$2.47  imes 10^3$	$1.62  imes 10^2$
	BSO	$2.55  imes 10^3$	$1.99  imes 10^4$	$5.92  imes 10^3$	$2.37  imes 10^3$
f22	LBSO	$2.32 \times 10^3$	$1.83  imes 10^4$	$5.20  imes 10^3$	$2.58  imes 10^3$
	NOBBSO	$2.31  imes 10^3$	$1.92  imes 10^4$	$5.49  imes 10^3$	$2.80  imes 10^3$
	BSO	$2.96  imes 10^3$	$1.28  imes 10^4$	$3.13  imes 10^3$	$2.66  imes 10^2$
f23	LBSO	$3.11  imes 10^3$	$1.27  imes 10^4$	$3.34  imes 10^3$	$2.89  imes 10^2$
	NOBBSO	$2.81  imes 10^3$	$1.36  imes 10^4$	$3.04  imes 10^3$	$2.79  imes 10^2$
	BSO	$3.07  imes 10^3$	$1.15  imes 10^4$	$3.35  imes 10^3$	$2.40  imes 10^2$
f24	LBSO	$3.13  imes 10^3$	$1.10  imes 10^4$	$3.38  imes 10^3$	$2.56  imes 10^2$
	NOBBSO	$3.03 \times 10^3$	$1.16 \times 10^4$	$3.19 \times 10^3$	$2.46 \times 10^{2}$
	BSO	$2.96 \times 10^{3}$	$3.22 \times 10^{5}$	$3.30 \times 10^3$	$6.07 \times 10^{3}$
f25	LBSO	$2.90 \times 10^3$	$2.88 \times 10^5$	$3.24 \times 10^3$	$6.08 \times 10^{3}$
	NOBBSO	$2.88 \times 10^3$	$2.45  imes 10^5$	$3.19  imes 10^3$	$5.84  imes 10^3$

Functions	Algorithms	Min	Max	Avg	Std
	BSO	$3.53 imes10^3$	$1.86 \times 10^5$	$7.51  imes 10^3$	$3.45  imes 10^3$
f26	LBSO	$2.87  imes 10^3$	$1.68 \times 10^5$	$7.08 \times 10^3$	$3.87  imes 10^3$
	NOBBSO	$2.83  imes 10^3$	$1.99 \times 10^5$	$7.23  imes 10^3$	$4.04  imes 10^3$
	BSO	$3.41  imes 10^3$	$4.24  imes 10^4$	$3.70  imes 10^3$	$6.96  imes 10^2$
f27	LBSO	$3.36  imes 10^3$	$2.67  imes 10^4$	$3.78  imes 10^3$	$6.37  imes 10^2$
	NOBBSO	$3.24  imes 10^3$	$3.04  imes 10^4$	$3.47  imes 10^3$	$6.25  imes 10^2$
f28	BSO	$3.32  imes 10^3$	$1.43  imes 10^5$	$3.64  imes 10^3$	$2.82 \times 10^3$
	LBSO	$3.21  imes 10^3$	$1.60 \times 10^5$	$3.49  imes 10^3$	$2.78  imes 10^3$
	NOBBSO	$3.18  imes 10^3$	$1.38 imes10^5$	$3.47  imes 10^3$	$2.67 imes10^3$
	BSO	$5.98  imes 10^5$	$8.30  imes 10^{11}$	$7.19 imes10^8$	$1.91  imes 10^{10}$
f30	LBSO	$3.27  imes 10^4$	$8.68  imes 10^{11}$	$9.47  imes 10^8$	$2.00  imes 10^{10}$
	NOBBSO	$1.29  imes 10^5$	$7.64 imes10^{11}$	$6.86  imes 10^8$	$1.88 imes10^{10}$
Statistica	al results	18/27	8/27	19/27	15/27

Table 4. Cont.

According to the evaluating indicator (Min) results of the experiments in Table 4, the designed NOBBSO algorithm has 18 functions that are more accurate than the original BSO and LBSO while only 8 functions that are worse. Moreover, from the evaluation indicator (Avg) it can be seen that the NOBBSO algorithm outperforms the original BSO algorithm and LBSO algorithm over 19 out of 27 functions while only lose on 8 functions. Hence it can be observed that the convergence accuracy of the BSO algorithm in terms of average value can be significantly improved with the help of the improved opposition-based learning technique. Therefore, on the whole, the upgraded algorithms have a larger chance of discovering optimal solutions than the other comparison algorithms. Meanwhile it also shows that the improved opposition-based learning technique has better success in identifying optimal solutions.

Furthermore, the optimization ability of NOBBSO on unimodal functions and multimodal functions is stronger than the original algorithm, but the optimization ability on hybrid functions and composition functions is slightly superior to other algorithms in some functions.

Nonetheless, in the light of the evaluating indicator (Std) it can be seen that compared with the original BSO algorithm and LBSO algorithm, the NOBBSO algorithm only has 15 functions that are better than other two algorithms and 12 ones that are poorer than other ones. Meanwhile, from the evaluating indicator (Max), we can see that the NOBBSO algorithm only has 8 better outcomes while 19 poorer results. From these two evaluation indexes, as can be seen that the proposed NOBBSO algorithm shows the poor stability in dealing with the test problems.

# 4.2.4. Another Accuracy Results of Different Comparison Algorithms

To have a fair comparison, another statistical analysis experiments are performed. Unlike the previous experiments, the dimension size of benchmark functions is 10 and the number of iterations is 500 and other conditions are the same. These are listed in Table 5.

Avg	Std
$5.26 \times 10^{9}$	$7.50  imes 10^{10}$
$4.70 imes10^9$	$7.34 imes10^{10}$
$3.44 \times 10^{19}$	$2.03  imes 10^{21}$

**Table 5.** Accurate results of two algorithms when D is 10.

Functions	Algorithms	Min	Max	Avg	Std
f1	BSO	$5.63  imes 10^2$	$1.96  imes 10^{12}$	$5.26 imes10^9$	$7.50  imes 10^{10}$
	NOBBSO	$2.12  imes 10^4$	$1.92  imes 10^{12}$	$4.70 imes10^9$	$7.34 imes10^{10}$
f2	BSO	$2.63  imes 10^3$	$6.80  imes 10^{22}$	$8.44  imes 10^{19}$	$2.03  imes 10^{21}$
	NOBBSO	$2.00  imes 10^2$	$6.62 imes10^{22}$	$6.04 imes10^{19}$	$1.57 imes10^{21}$
f3	BSO	$3.10  imes 10^2$	$9.59  imes 10^{12}$	$1.09  imes 10^{10}$	$2.69  imes 10^{11}$
	NOBBSO	$3.00  imes 10^2$	$1.93  imes 10^{13}$	$1.90  imes 10^{10}$	$4.73  imes 10^{11}$
f4	BSO	$4.00  imes 10^2$	$1.32  imes 10^5$	$6.29  imes 10^2$	$4.52  imes 10^3$
	NOBBSO	$4.00  imes 10^2$	$1.75  imes 10^5$	$6.47  imes 10^2$	$5.09  imes 10^3$
f5	BSO	$5.07  imes 10^2$	$1.28  imes 10^3$	$5.31  imes 10^2$	$3.40  imes 10^1$
	NOBBSO	$5.07  imes 10^2$	$1.25 imes10^3$	$5.26  imes 10^2$	$3.52  imes 10^1$
f6	BSO	$6.16  imes 10^2$	$1.15  imes 10^3$	$6.37  imes 10^2$	$2.16  imes 10^1$
	NOBBSO	$6.05  imes 10^2$	$1.11 imes10^3$	$6.19 imes10^2$	$2.21  imes 10^1$
f7	BSO	$7.23 \times 10^2$	$2.73 \times 10^3$	$7.63  imes 10^2$	$8.88  imes 10^1$
	NOBBSO	$7.25 \times 10^2$	$2.83  imes 10^3$	$7.42  imes 10^2$	$8.75  imes 10^1$
f8	BSO	$8.09 \times 10^{2}$	$1.29  imes 10^3$	$8.23  imes 10^2$	$2.41  imes 10^1$
	NOBBSO	$8.07  imes 10^2$	$1.30 \times 10^3$	$8.21  imes 10^2$	$2.34  imes 10^1$
f9	BSO	$9.42  imes 10^2$	$8.74 imes10^4$	$1.23  imes 10^3$	$2.68  imes 10^3$
	NOBBSO	$9.04 imes10^2$	$6.62 imes10^4$	$1.10  imes 10^3$	$2.36  imes 10^3$
f10	BSO	$1.53  imes 10^3$	$7.04  imes 10^3$	$2.22 \times 10^3$	$4.73  imes 10^2$
	NOBBSO	$1.39  imes 10^3$	$7.03  imes 10^3$	$2.14  imes 10^3$	$5.41 \times 10^2$
f11	BSO	$1.11 \times 10^3$	$1.54 imes10^{10}$	$2.45  imes 10^7$	$5.56  imes 10^8$
	NOBBSO	$1.11 \times 10^3$	$1.44 imes10^{10}$	$2.29  imes 10^7$	$5.23  imes 10^8$
f12	BSO	$5.71 \times 10^3$	$4.81  imes 10^{11}$	$7.93  imes 10^8$	$1.68  imes 10^{10}$
	NOBBSO	$3.78  imes 10^3$	$3.95  imes 10^{11}$	$7.18  imes 10^8$	$1.51 imes10^{10}$
f13	BSO	$2.33  imes 10^3$	$3.55  imes 10^{11}$	$6.01  imes 10^8$	$1.33  imes 10^{10}$
	NOBBSO	$2.48 \times 10^3$	$3.49 imes10^{11}$	$6.07  imes 10^8$	$1.35  imes 10^{10}$
f14	BSO	$1.49  imes 10^3$	$2.88  imes 10^{10}$	$4.48  imes 10^7$	$1.01 \times 10^{9}$
	NOBBSO	$1.49  imes 10^3$	$2.89  imes 10^{10}$	$4.89 \times 10^{7}$	$1.10 \times 10^{9}$
f15	BSO	$1.58  imes 10^3$	$5.40  imes 10^{11}$	$8.96  imes 10^8$	$2.03  imes 10^{10}$
	NOBBSO	$1.75 \times 10^3$	$4.98 imes10^{11}$	$8.58 imes10^8$	$1.93 imes10^{10}$
f16	BSO	$1.62 \times 10^3$	$1.74  imes 10^5$	$2.08  imes 10^3$	$5.27  imes 10^3$
	NOBBSO	$1.60  imes 10^3$	$1.96 \times 10^{5}$	$2.04 \times 10^{3}$	$6.49 \times 10^{3}$
f18	BSO	$2.05  imes 10^3$	$1.26 \times 10^{11}$	$2.08  imes 10^8$	$4.64  imes 10^9$
	NOBBSO	$2.05 \times 10^3$	$1.17  imes 10^{11}$	$1.95  imes 10^8$	$4.37  imes 10^9$
f19	BSO	$1.93 \times 10^3$	$1.08  imes 10^{12}$	$1.77 \times 10^9$	$3.99  imes 10^{10}$
	NOBBSO	$1.99  imes 10^3$	$9.87 imes10^{11}$	$1.69 imes10^9$	$3.82  imes 10^{10}$
f21	BSO	$2.20 \times 10^3$	$1.20  imes 10^4$	$2.24  imes 10^3$	$2.75  imes 10^2$
	NOBBSO	$2.20 \times 10^3$	$1.21 \times 10^4$	$2.30  imes 10^3$	$2.55  imes 10^2$
f22	BSO	$2.30  imes 10^3$	$8.02 \times 10^3$	$2.35  imes 10^3$	$2.81  imes 10^2$
1	NOBBSO	$2.30  imes 10^3$	$9.12  imes 10^3$	$2.33  imes 10^3$	$2.78  imes 10^2$

Functions	Algorithms	Min	Max	Avg	Std
f23	BSO	$2.62 \times 10^3$	$7.25 \times 10^3$	$2.68  imes 10^3$	$1.85  imes 10^2$
120	NOBBSO	$2.61  imes 10^3$	$7.08  imes 10^3$	$2.64 imes10^3$	$1.73  imes 10^2$
f24	BSO	$2.50  imes 10^3$	$8.79 \times 10^3$	$2.75  imes 10^3$	$2.23  imes 10^2$
121	NOBBSO	$2.50  imes 10^3$	$7.29 imes10^3$	$2.71  imes 10^3$	$2.06 imes10^2$
f25	BSO	$2.90  imes 10^3$	$3.88  imes 10^4$	$3.01 \times 10^3$	$1.37  imes 10^3$
120	NOBBSO	$2.90  imes 10^3$	$4.33  imes 10^4$	$2.99  imes 10^3$	$1.39  imes 10^3$
f26	BSO	$2.80  imes 10^3$	$2.93  imes 10^4$	$3.23  imes 10^3$	$8.67  imes 10^2$
	NOBBSO	$2.90  imes 10^3$	$3.03  imes 10^4$	$3.04 imes10^3$	$8.87  imes 10^2$
f27	BSO	$3.11  imes 10^3$	$2.50  imes 10^4$	$3.18  imes 10^3$	$7.47  imes 10^2$
127	NOBBSO	$3.10 imes10^3$	$2.76  imes 10^4$	$3.16 imes10^3$	$8.88  imes 10^2$
f28	BSO	$3.11  imes 10^3$	$2.68  imes 10^4$	$3.38 \times 10^3$	$8.06 \times 10^2$
120	NOBBSO	$3.17  imes 10^3$	$1.69 imes10^4$	$3.38 \times 10^3$	$4.77  imes 10^2$
f30	BSO	$4.22  imes 10^3$	$2.50  imes 10^{11}$	$3.51  imes 10^8$	$7.96  imes 10^9$
150	NOBBSO	$1.24  imes 10^4$	$2.34 imes10^{11}$	$3.43 imes10^8$	$7.88 imes10^9$
Statistica	al results	10/27	16/27	21/27	16/27

Table 5. Cont.

Compared with the original algorithm, NOBBSO obtains more accurate solutions on 10 functions. It is worth noting that there are nine functions that show that they have the same minimum value. The analysis shows that the optimization ability of the NOBBSO is stronger than the original algorithm. Moreover, the evaluation indicator (Avg) shows the optimal solution obtained by NOBBSO is generally more accurate than the original algorithm.

## 4.2.5. Convergence Analysis

Figure 3 represents the convergent curves of 27 CEC2017 benchmark functions that are performed on the above three BSO algorithms, where the scale of the *y*-axis is logarithmic.

The convergent curves of most functions show that NOBBSO obtains faster convergent speed than the classic BSO algorithm. Specifically, the analysis of the slopes of most functions clearly shows that the slope of NOBSO is better than the other algorithms, which suggests that NOBBSO has faster convergence speed than BSO, and also shows that it has the ability to search for the optimal solution more quickly.

However, unfortunately, for few functions such as f13, f15, f30, the convergence speed obtained by the NOBBSO algorithm is slower than the other algorithm, which means NOBBSO is attracted to locally optimal solutions, which results in its inability to find a better solution.



Figure 3. Cont.



Figure 3. The analysis diagram of convergence.

## 4.2.6. Friedman Test

A statistical mathematical analysis method, namely the Friedman test proposed by M. Friedman in 1973 [21], is used to reflect the differences among multiple samples. Average ranking is an index to evaluate the difference between samples in Friedman test. The smaller the value is, the greater the difference among samples is. For algorithms, the smaller the value is, it shows that there is a significant difference between the designed algorithm and the algorithm involved in the comparison within confidence level, that is, the performance of the designed algorithm is better than other algorithms because the average ranking is calculated by average values.

Table 6 lists the statistical results of Friedman test of three algorithms when the confidence level is 0.05.

Table 6. Statistical results.

Algorithms	Average Ranking
BSO	2.52
LBSO	2.41
NOBBSO	1.42

We can see from the Table 6 that NOBBSO has obtained a minimum value that the average ranking represents, which demonstrates that the NOBBSO algorithm is different from other algorithms significantly.

# 5. Discussions

The numerical experiment results demonstrate that the NOBBSO algorithm has higher convergent speed and accuracy in comparison with other competitors.

However, NOBBSO is not a perfect algorithm either because its stability is not very good. For example, Table 3 shows that the  $\lambda$  parameter is continuously changed while the evaluation indicator (Std) does not change much. Therefore, the adaptive strategy is not the main reason for the impact on its stability and the fact reason is mainly from novel opposition-based learning. Because the new solution generated has also certain randomness, which can have an impact on its stability. Moreover, by analyzing Tables 4 and 5, they show that NOBBSO is not much different from the original algorithm in obtaining the optimal solution in dealing with low dimension benchmark functions while NOBBSO is more accurate in obtaining the optimal solution in high dimensions, which demonstrates that the NOBBSO algorithm has advantages in dealing with higher dimensional problems.

From the worse results of few benchmark functions in Tables 4 and 5, NOBBSO still has a certain probability to fall into the local optimal solution, but the probability of falling into a local optimum is actually less than the original algorithm on the whole. Hence it is still worth looking for better strategy or algorithms to improve it.

## 6. Conclusions

A novel algorithm called adaptive beetle swarm algorithm with novel oppositionbased learning is proposed in this study. The algorithm is on the basis of the novel opposition-based learning and adaptive strategy to further enhance the optimization performance of the BSO algorithm. By testing 27 CEC2017 functions, the results show that the proposed NOBBSO algorithm has higher convergence accuracy and convergence speed.

In the future, NOBBSO can still be used in many places, because it is easy to converge and converges quickly. In engineering, there are many complex optimization problems that need to be solved well, but they are not quickly solved by traditional methods. Therefore, NOBBSO can be applied to solve them, which is also the future research work.

Author Contributions: Conceptualization, Q.W. and P.S.; methodology, Q.W. and P.S.; software, Q.W.; validation, Q.W., G.C. and P.S.; formal analysis, G.C. and P.S.; investigation, P.S.; resources, Q.W. and P.S.; data curation, Q.W. and P.S.; writing—original draft preparation, Q.W. and G.C.; writing—review

and editing, Q.W. and P.S.; visualization, Q.W.; supervision, P.S.; project administration, P.S.; funding acquisition, P.S. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work is funded by the Technology Plan Projects of Jiangxi Provincial Education Department (No.GJJ200424).

Data Availability Statement: There is no data availability.

Conflicts of Interest: The authors declare that they have no known competing financial interests.

## References

- 1. Kennedy, J.; Eberhart, R. Particle Swarm Optimization. In Proceedings of the ICNN95-international Conference on Neural Networks, Perth, Australia, 27 November–1 December 1995.
- Karaboga, D.; Akay, B. A comparative study of Artificial Bee Colony algorithm. *Appl. Math. Comput.* 2009, 214, 108–132. [CrossRef]
- 3. Parpinelli, R.S.; Lopes, H.S.; Freitas, A.A. Data mining with an ant colony optimization algorithm. *Evol. Comput. IEEE Trans.* 2002, *6*, 321–332. [CrossRef]
- 4. Wang, T.; Long, Y.; Qiang, L. Beetle Swarm Optimization Algorithm: Theory and Application. *arXiv* **2018**, arXiv:1808.00206. [CrossRef]
- Wang, L.; Wu, Q.; Lin, F.; Li, S.; Chen, D. A New Trajectory-Planning Beetle Swarm Optimization Algorithm for Trajectory Planning of Robot Manipulators. *IEEE Access* 2019, 7, 154331–154345. [CrossRef]
- 6. Hariharan, B.; Siva, R.; Kaliraj, S.; Prakash, P.N. ABSO: An energy-efficient multi-objective VM consolidation using adaptive beetle swarm optimization on cloud environment. *J. Ambient. Intell. Humaniz. Comput.* **2021**, 1–13. [CrossRef]
- Mu, Y.; Li, B.; An, D.; Wei, Y. Three-Dimensional Route Planning Based on the Beetle Swarm Optimization Algorithm. *IEEE Access* 2019, 7, 117804–117813. [CrossRef]
- 8. Singh, P.; Kaur, A.; Batth, R.S.; Kaur, S.; Gianini, G. Multi-disease big data analysis using beetle swarm optimization and an adaptive neuro-fuzzy inference system. *Neural Comput. Appl.* **2021**, *33*, 10403–10414. [CrossRef]
- 9. Jiang, Y.; Wang, S.; Li, Y. Localizing and quantifying structural damage by means of a beetle swarm optimization algorithm. *Adv. Struct. Eng.* **2020**, *24*, 136943322095682. [CrossRef]
- 10. Zhou, L.; Chen, K.; Dong, H.; Chi, S.; Chen, Z. An Improved beetle swarm optimization Algorithm for the Intelligent Navigation Control of Autonomous Sailing Robots. *IEEE Access* 2020, *9*, 5296–5311. [CrossRef]
- 11. Zhang, P.; Zhang, J.; Elsabbagh, A. Gait multi-objectives optimization of lower limb exoskeleton robot based on BSO-EOLLFF algorithm. *Robotica* 2022, 1–19. [CrossRef]
- Tizhoosh, H.R. Opposition-Based Learning: A New Scheme for Machine Intelligence. In Proceedings of the International Conference on International Conference on Computational Intelligence for Modelling, Control & Automation, Vienna, Austria, 28–30 November 2005; pp. 695–701.
- 13. Wang, H.; Wu, Z.; Rahnamayan, S.; Liu, Y.; Ventresca, M. Enhancing particle swarm optimization using generalized oppositionbased learning. *Inf. Sci.* **2011**, *181*, 4699–4714. [CrossRef]
- 14. Shao, P.; Wu, Z.J.; Zhou, X.Y.; Deng, C.S. Improved Particle Swarm Optimization Algorithm Based on Opposite Learning of Refraction. *Acta Electron. Sin.* **2015**, *25*, 4117–4125.
- 15. Zhou, X.Y.; Wu, Z.J.; Wang, H.; Li, K.S.; Zhang, H.Y. Elite Opposition-Based Particle Swarm Optimization. *Acta Electron. Sin.* **2013**, 41, 1647–1652.
- 16. Qian, Q.; Deng, Y.; Sun, H.; Pan, J.; Yin, J.; Feng, Y.; Fu, Y.; Li, Y. Enhanced beetle antennae search algorithm for complex and unbiased optimization. *Soft Comput.* **2022**, *26*, 10331–10369. [CrossRef] [PubMed]
- 17. Zhao, J.; Lv, L.; Fan, T.; Wang, H.; Li, C.; Fu, P. Particle swarm optimization using elite opposition-based learning and application in wireless sensor network. *Sens. Lett.* **2014**, *12*, 404–408. [CrossRef]
- 18. Mohammadzadeh, A.; Taghavifar, H. A robust fuzzy control approach for path-following control of autonomous vehicles. *Soft Comput.* **2020**, *24*, 3223–3235. [CrossRef]
- 19. Jiang, X.; Li, S. BAS: Beetle Antennae Search Algorithm for Optimization Problems. arXiv 2017, arXiv:1710.10724. [CrossRef]
- Awad, N.H.; Ali, M.Z.; Suganthan, P.N.; Liang, J.J.; Qu, B.Y. Problem Definitions and Evaluation Criteria for the CEC 2017 Special Session and Competition on Single Objective Bound Constrained Real-Parameter Numerical Optimization; Nanyang Technological University Singapore: Singapore, 2016.
- 21. Friedman, M. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *J. Am. Stat. Assoc.* **1937**, 32, 675–701. [CrossRef]