IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received August 31, 2021, accepted September 23, 2021, date of publication October 4, 2021, date of current version October 13, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3117567

An Improved Beetle Antennae Search Algorithm Based on the Elite Selection Mechanism and the Neighbor Mobility Strategy for Global Optimization Problems

XUAN SHAO AND YUQI FAN
Department of Electrical Engineering, College of Rongcheng, Harbin University of Science and Technology, Rongcheng 264300, China
Corresponding author: Xuan Shao (xuanshao84@163.com)

This work was supported by the International Cooperation Projects under Grant 2012DFR70840.

ABSTRACT Aiming at the shortcoming that the basic beetle antennae search algorithm fails to consider differences between individuals and the dynamic information in the searching process, this paper proposed a new beetle antennae search algorithm based on the elite selection mechanism and the neighbor mobility strategy. The elite selection mechanism will be used to weaken beetles having bad performances and generate new beetles to ensure diversities and abilities in the whole population. The neighbor mobility strategy will guide the algorithm to open up a wider searching area to ensure that individuals having good positions own a chance to infect individuals with poor performances. To verify the searching ability and the optimization speed of the proposed algorithm in this paper, different testing functions were selected for numerical testing experiments, and the iteration figures, box plots, and searching path figures were given. The experimental results showed that the proposed algorithm in this paper was superior to the original algorithm in the solving accuracy, the convergence speed, and the stability.

INDEX TERMS Beetle antennae search algorithm, numerical problems, global optimization.

I. INTRODUCTION

In practical engineering fields, many optimization problems need to be solved under complex constraints and in a large searching range. Traditional mathematical methods, such as the steepest descent method and the variable scale method, can only calculate simple and continuous functions [1]–[3]. Therefore, for nonlinear, multivariable, multi-constraint, multi-dimension, and other complex optimization problems, engineering fields need a strong computing power and a high precision optimization strategy to solve the different problems [4]–[6]. The swarm intelligence optimization is inspired by the biology, which is proposed by bionics and mathematicians on the swarm intelligence theory. Its central idea draws and simulates the cluster phenomenon and the behavior in natural creatures [7], [8]. The swarm intelligence optimization algorithm is to achieve some complex tasks by individuals in the biological population without the centralized control guidance and the simple cooperation. Therefore, the swarm intelligence optimization algorithm can solve complex optimization problems without the global information. In recent years, due to the improvement of practicabilities and fault tolerances, swarm intelligence optimization algorithms have been widely used in different fields [9]–[11]. In recent years, scholars have proposed many advanced metaheuristic algorithms, such as archimedes optimization algorithm (AOA) [12], bald eagle search algorithm (BES) [13], harris hawks optimizer (HHO) [14], socio evolution and learning optimization (SELO) [15], sooty tern optimization algorithm (STOA) [16], spotted hyena optimizer (SHO) [17], teaching learning based optimization (TLBO) [18].

Beetle Antennae Search Algorithm (BAS) is an efficient swarm intelligence algorithm proposed by Xiangyuan Jiang and Shuai Li in 2017 [19]. It is an evolutionary algorithm that imitates the beetle evolution behavior, the foraging behavior, and the courtship behavior. The beetle survival behavior in nature can be modeled as the algorithm optimization

The associate editor coordinating the review of this manuscript and approving it for publication was Geng-Ming Jiang.

process for the fitness function, and the searching target can be modeled as the optimal solution in the fitness function. The algorithm not only has a strong ability of the individual identification and the environment identification, but also does not need to know the function gradient information. In addition, its program code is simple and can quickly obtain the optimal solution under the condition of a stable convergence [20]–[23]. At present, it has been successfully applied to a variety of industrial engineering optimization problems, and has a high potential research value [24]–[29]. Ameer Tamoor Khan *et al.* proposed the quantum beetle antennae search algorithm and applied it in the constrain portfolio problem [30]. Literature [31] used BAS for the UAV sensing and the avoidance of obstacles. Aghila Rajagopal *et al.* proposed a new hybrid extreme learning machine with beetle antennae search algorithm, and used BAS to evaluate the optimal strategy in the low earth orbit satellite communication networks [32]. Meijin Lin *et al.* used BAS in the economic load distribution problem of power systems [33]. Literature [34] devised a new fallback BAS for the path planning. Jiang *et al.* combined the BAS with non-trivial mechanisms to solve the 3-D path planning problem [35]. Shuo Xie *et al.* proposed an improved Q-learning BAS to solve the model predictive ship collision avoidance [36]. Shuzhi Gao *et al.* designed a chaos BAS for neural network soft-sensor model of the PVC Polymerization Process [37]. Vasilios N. Katsikis *et al.* studied the time-varying minimum cost portfolio insurance problem used the BAS [38]. Qing Wu *et al.* introduced a neural network classifier based the BAS for the pattern classification [39]. Jianhui Yang and Zhenrui Peng designed the beetle-swarm evolution competitive algorithm for bridge sensor optimal placements [40]. Lin Zhou used an improved beetle swarm optimization algorithm for autonomous sailing robots [41]. Xin Xu used lévy flights and adaptive strategythe in the BAS [42]. Lei Wang drafted a new BAS for the trajectory planning of robot manipulators [43]. Yaozong Cheng proposed a motion planning of the redundant manipulator based on the BAS [44]. Literature [45] introduced a model free approach for the online optimization using the BAS. Zongcheng Yue *et al.* presented a hardware descriptive approach based the BAS [46]. Tamal Ghosh and Kristian Martinsen proposed a collaborative the BAS memory based on the adaptive Learning [47]. Ameer Tamoor Khan *et al.* showed an improved BAS with Zeroing Neural Network for the online solution of the constrained optimization [48]. Literature [49] designed a BAS dynamical attitude configuration with wearable wireless body sensor networks. Literature [50] devised a support vector machine for wind turbine rolling bearings fault diagnosis under the BAS. Heng Zhang *et al.* invented a beetle colony optimization algorithm [51]. Literature [52] displayed the trajectory optimization used the BAS. Zhiqiang Liao *et al.* introduced an automatic bearing fault feature extraction method based on the BAS [53]. Jiandong Huang *et al.* proposed an accurately predicting dynamic modulus of asphalt mixtures in low temperature regions based the BAS [54]. Compared with other classical algorithms, BAS also owns a competitiveness. For GA, BAS has a less complexity because of no-binary coding. For PSO, BAS has a large jumping speed. For FA, BAS has less initial parameters. For artificial bee colony (ABC), BAS has a larger searching ability and a lower complexity [55], [56].

Because BAS is a fixed-reduction step algorithm based on the natural selection, its attenuation factor and initial step are fixed, which helps to maintain the algorithm stability. However, the changing of the exploration field and the searching step are fixed in the algorithm implementation process, and there is no significant differences in the early and late searching stages, which makes the local searching not thorough, and leads to a low precision searching result. The initial step is selected according to the artificial experience. A lot of experiments must be done before the artificial experience selection whose process is complex, time-consuming, and laborious. And in engineering applications, the parameter selection process must consider working environments, errors, interferences, and other factors, so the optimal initialization parameters of the algorithm can not be obtained by artificial experiences. Therefore, to fully improve the algorithm performance, it is necessary to design an enhanced way to meet the expectation field and step changes. This paper proposed an improved algorithm based on the elite selection mechanism and the neighbor mobility strategy (ENBAS). In ENBAS, the exponential equation factor was added to the basic BAS, and a new iteration updated mechanism was added in the global searching and local searching stages to increase the population diversity, which can improve the algorithm convergence speed. Benchmark functions were used for function experiments, and this paper compared the proposed algorithm with other optimization algorithms to verify the ENBAS performance.

II. BEETLE ANTENNAE SEARCH ALGORITHM

The BAS physiological principle is that two antennae of the beetle will receive different intensities of food odor pheromones when the beetle is in an unknown position. And the beetle will determine its movement direction according to the pheromone strength received by its antennae. When the pheromone received by the left antennae is stronger than that of the right, the beetle will move to the left, on the contrary, it will move to the right. In BAS, the optimal value of the fitness function to be found is regarded as the food that the beetle seeks in nature, and the independent variable of the fitness function is regarded as the beetle position in the searching space. To explore the initial unknown environment, it is assumed that the initial searching direction is any direction in any dimension space. Therefore, the BAS random searching direction can be standardized as follows:

$$\vec{b} = b = \frac{rnd(dim, 1)}{\|rnd(dim, 1)\|} \tag{1}$$

In (1), *rnd* represents the random vector function and *dim* represents the searching dimension. Because the beetle does not know the exact food position in nature, the beetle uses two antennae to detect the food pheromone and then judges the next step direction. The position coordinates of two antennae can be expressed as follows:

$$x_r = x^t + d^t \cdot b \tag{2}$$

$$x_l = x^t - d^t \cdot b \tag{3}$$

In equations (2) and (3), t is the number of iterations, x_r is the right antennae position, x_l is the left antennae position, x^t is each beetle position in the searching space, and d^t is the detection distance of the beetle. If the left antennae receive a stronger odor than that of the left antennae, it will move to the left side, otherwise, it will move to the right side. BAS can update the beetle position by judging the intensity difference between two antennae. So the next beetle step is as follows:

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

In equation (4), t is the number of iterations, δ^t is the searching step. *Sign* is a sign function.

$$sign\left(f\left(x_r\right) - f\left(x_l\right)\right) = \begin{cases} 1, & f\left(x_r\right) - f\left(x_l\right) > 0 \\ 0, & f\left(x_r\right) - f\left(x_l\right) = 0 \\ -1, & f\left(x_r\right) - f\left(x_l\right) < 0 \end{cases} \tag{5}$$

The detection distance d^t and searching step δ^t can be updated by:

$$d^{t+1} = 0.95d^t + 0.01 \tag{6}$$

$$\delta^{t+1} = 0.95\delta^t \tag{7}$$

To explain BAS more clearly, the paper gives the BAS pseudo code.

Algorithm 1 BAS

Input: *Fitness function* $F(\cdot)$. $t_{\max}$, N *beetles positions.* $t = 1$. *All initial parameters. Searching range. Initial optimum solution* x_{best}. *Initial optimum value* g_{best}.
Output: x_{best}, g_{best}.

```
While ($t < t_{\max}$)
    For $i = 1 : N$
        Define $b$
        $x_r = x_i^t + d^t . b$
        $x_l = x_i^t - d^t . b$
        $x_i^{t+1} = x_i^t + \delta^t b sign(f(x_r) - f(x_l))$
    End For
    For $i = 1 : N$
        If $F(x_i^{t+1})$ is better than $g_{best}$
            $x_{best} = x_i^{t+1}$
            $g_{best} = F(x_i^{t+1})$
        End If
    End For
    $d^{t+1} = 0.95\ d^t + 0.01$
    $\delta^{t+1} = 0.95\ \delta^t$
    $t = t + 1$
End While
```

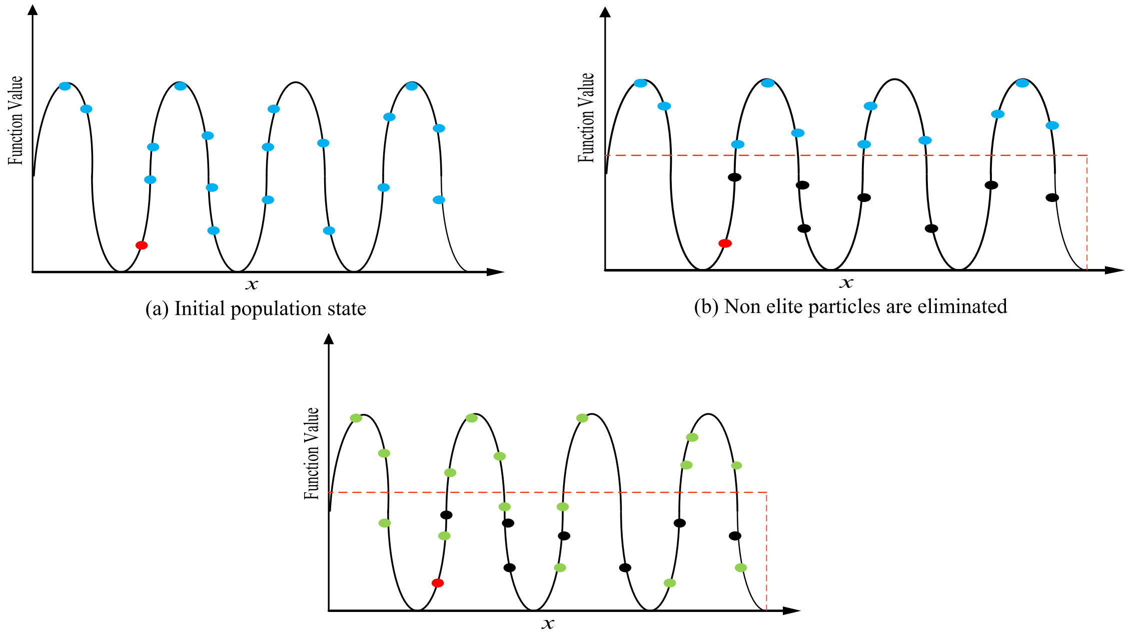
III. AN IMPROVED BEETLE ANTENNAE SEARCH ALGORITHM

A. THE ELITE SELECTION MECHANISM

The elite selection mechanism is to find the elite individual with the good convergence and the robustness in the whole algorithm population, which can guide other individuals to search for a better solution [42], [57]. After a certain number of iterations, all individual positions in the whole population are sorted according to function values calculated by all individuals, and the best position x^* is found. Then, establish the ***R*** mathematical set to save all elite particles. Because the ***R*** mathematical set is empty and x^* has the best fitness value, and the best individual position x^* will be directly added to the ***R*** mathematical set, then, the Euclidean distance (g^*, g) between the individual fitness value g and the optimal individual fitness value g^* will be calculated, and the threshold value E is set. If the Euclidean distance (g^*, g) is less than E, the distribution and convergence of the current particle can be regarded as the elite particle and will be entered into the elite set ***R***. If the Euclidean distance (g^*, g) is larger than E, it can be regarded as a non-elite individual, and will be eliminated, then, a new individual will be randomly generated to replace the non-elite. Figure 1 illustrates the elite selection mechanism principle. The abscissa is the individual position, and the ordinate is the fitness value in Figure 1. It must be mentioned here that this paper takes the minimum value in one-dimensional function searching process as an example. Figure 1(a) shows the algorithm initial state when the proposed algorithm does not operate the elite selection mechanism. The red particle is the best position in the whole population. In other words, the fitness value of the red particle is the smallest. Then, the Euclidean distance (g^*, g) between the fitness values of each blue particle and that of the red particle will be calculated, and all distances that are less than the threshold value E will be added to the elite set ***R*** in the fitness descending order. In Figure 1(b), the elite particles in the ***R*** set are all black particles under the red dotted line. All the blue non-elite particles in Figure 1(b) are eliminated, and finally, new individuals will be randomly generated to replace all sifted-out non-elite particles. In Figure 1(c), green particles are new individuals generated after eliminating non-elite particles. It can be seen from Figure 1(c) that some green particles are in the ***R*** set, which indicates that the elite selection mechanism avoids the large-scale reproduction and covering the whole population of bad individuals in the later iteration stage.

B. THE NEIGHBOR MOBILITY STRATEGY

The neighbor mobility strategy is to exchange information among individuals, to learn from other individuals experiences to improve their movement direction diversities, and achieve the information sharing purpose [57]. When the algorithm is running, the neighboring individuals will form a local small group to quickly find the global optimal solution. Each small group generates a local optimum individual to

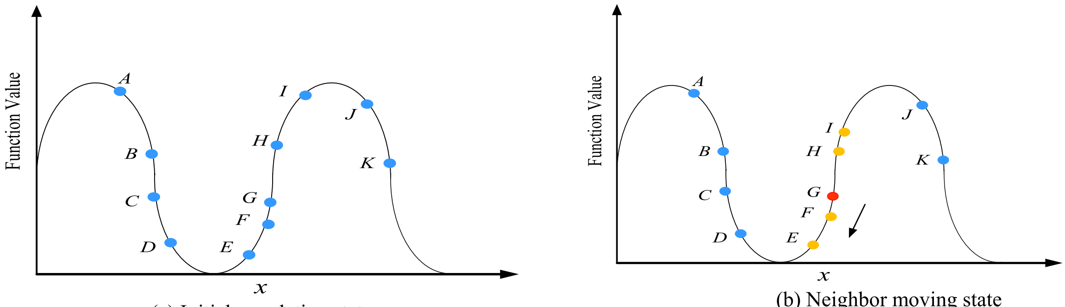


(a) Initial population state

(b) Non elite particles are eliminated

(c) Individuals are randomly generated

FIGURE 1. The flow chart of the elite selection mechanism.



(a) Initial population state

(b) Neighbor moving state

FIGURE 2. The flow chart of the neighbor mobility strategy.

guide other individuals in the group to search for the global optimum solution in a better direction. The information between different groups will be exchanged to achieve the information sharing after the algorithm runs repeatedly, which effectively avoids the premature convergence, and enables all individuals in the algorithm to search for the better solution in the correct direction. The neighbor mobility strategy is that each individual finds and moves to M particles which are closest to its Euclidean distance, to achieve the evolution goal. The gait size formula is as follows:

$$c = x_{best} + x_{best} \cdot rands \cdot (t/t_{\max}) \tag{8}$$

where x_{best} is the optimum solution, and *rands* is randomly selected in the range of $[-1, 1]$. At the beginning of the searching process, the initial size is large. With the iteration process, t increases, t_{max} remains unchanged, t/t_{max} gradually increases, thus, the range of moving size gradually increases, which ensures that the ENBAS can jump out of the local extremum. Figure 2 is the schematic diagram of the neighbor mobility strategy. The abscissa is the individual position, and the ordinate is the fitness value in Figure 2. It must be mentioned here that this paper takes the minimum value in one-dimensional function searching process as an example. Taking particle G as the example, this paper firstly finds the four particles $EFHI$ which are closest to G in the searching space and then calculates fitness values of four particles. Because it is a searching minimum problem, the function value of the particle E is the smallest. Particle G moves to particle E, and the arrow represents the particle G direction. The neighbor mobility strategy combines the global searching with the local depth searching so that some particles having excellent performances can be transferred between different small groups.

C. THE PROPOSED ALGORITHM

There are two strategies in ENBAS, including the elite selection mechanism and the neighbor mobility strategy. Through the two strategies, ENBAS can find a better solution. The elite selection mechanism can eliminate poor performances of particles and generate new particles in each iteration, which can ensure the diversity and the adaptability in the whole population. The neighbor mobility strategy can effectively guide the algorithm to search the better direction according to the local optimal information, and ensure that some individuals with excellent performances in the population having a greater chance to infect other individuals with poor performances. In ENBAS first runs the elite selection mechanism to eliminate and update some individuals with poor performances, and the updated individuals will return to their group for the next iteration. After running the elite selection mechanism, ENBAS will divide all beetles into several parallel groups, and allow particles in each group to search and evolve, and to complete the updated operation of the population position, and make each beetle is closer to the searching target. ENBAS can effectively open up a new searching area, thus improving the global searching ability to jump out of the local area. In ENBAS, to make all individuals are closer to the global optimal solution and avoid the inaccuracy of artificial selection, two updating formulas of beetle antennal positions will be changed to:

$$x_{rnew} = x^t + d^t \cdot u_1 \tag{9}$$

$$x_{lnew} = x^t - d^t \cdot u_2 \tag{10}$$

where u_1 and u_2 are random numbers with normal Gaussian distribution.

The new location update formula is

$$x_{new}^{t+1} = x_{best} - x_{best} \cdot u_3 \cdot b \cdot sign\left(f\left(x_{rnew}\right) - f\left(x_{lnew}\right)\right) d^t \cdot u_2 \tag{11}$$

where u_3 is random numbers with normal Gaussian distribution. In this way, the left-right antennae will not blindly seek the optimal solution in the searching space, and gradually move along the direction of the optimal solution. The normal Gaussian distribution is used in ENBAS to replace the detection distance and the searching step, which reduces the running time and provides the running efficiency. After updated strategies, new individuals will have a better searching ability, which improves the whole algorithm performance. To explain ENBAS more clearly, the paper gives the ENBAS pseudo code.

Algorithm 2 ENBAS

```
Input: Fitness function $F(\cdot)$. $t_{\max}$, $N$ beetles positions. $t = 1$. All initial parameters. Searching range. Initial optimum solution $x_{best}$. Initial optimum value $g_{best}$.
Output: $x_{best}$, $g_{best}$.

While ($t < t_{\max}$)
    For $i = 1$: $N$
    Define $b$
    $x_{rnew} = x_i^t + d^t \cdot u_1$
    $x_{lnew} = x_i^t - d^t \cdot u_2$
    $x_{newi}^{t+1} = x_{best} - x_{best} u_3 \cdot b\ sign(f(x_{rnew}) - f(x_{lnew}))$
        If $F(x_{newi}^{t+1})$ is better than $F(x_{best})$
            $x_{best} = x_{newi}^{t+1}$
            $g_{best} = F(x_{newi}^{t+1})$
        End If
    End For
    Elite selection mechanism
    Neighbor mobility strategy
    $c = x_{best} + x_{best}\ rands \cdot (t/t_{max})$
      t = t + 1
End While
```

The ENBAS implementation steps as follows:

Step 1: Randomly generate N beetle positions. Set $t = 1$ and the maximum iteration t_{max}. The initial detection distance is set to the upper limit of the searching range. The fitness function is determined and the beetle movement direction is set according to the optimization purpose of the fitness function. All beetle positions are brought to the fitness function for the selection experiment. After testing, the initial optimum solution is selected and defined as the initial global optimum solution, and saved for the next generation.

Step 2: Update the random searching direction using the formula (1) in D-dimension searching space. Use formulas (9) and (10) to update beetle antennae positions in this iteration. Finally, the position coordinates of the left-right antennae are brought into the function to calculate the fitness value, and the difference between the two antennae is calculated.

Step 3: Update all beetle positions using the formula (11).

Step 4: Executive the elite selection mechanism and the neighbor mobility strategy.

Step 5: Calculate and compare all fitness values to determine the current optimum fitness value and solution. Select and update the global optimum fitness value and solution.

Step 6: Set $t = t + 1$, and judge whether the iteration number meets the iteration termination condition $t = t_{max}$. If t does not satisfy the termination condition, return to Step 2 and start the next iteration cycle until $t = t_{max}$.

D. THE COMPUTATIONAL COMPLEXITY

The algorithm complexity can be divided into the time complexity and the space complexity. The space complexity refers to the memory space needed to execute the algorithm. The algorithm time complexity is a function, which qualitatively displays the algorithm running time. The time complexity is usually expressed by large $\boldsymbol{O}$ symbol, excluding the low order term and the first term coefficient in the function. In this way, the time complexity can be said to be asymptotic when the input value approaches infinity. Initialization of ENBAS population needs $\boldsymbol{O}(N \times D)$ time where N indicates the number of iterations to generate random population in a test function. In the next step, the function fitness of each agent requires $\boldsymbol{O}\ (T_{max} \times N \times D)$ time where T_{max} is the maximum iteration number. It requires $\boldsymbol{O}\ (T_{max} \times N)$ time to define the group of ENBAS.

TABLE 1. Basic information of benchmark functions.

Name	Formulation	Scope$_{(D=2)}$	Scope$_{(D=50,100)}$	A_{im}
Alpine-01	$f_1(x)=\sum_{i=1}^{D}\lvert x_i\sin(x_i)+0.1x_i\rvert$	[-10, 10]	[-1, 1]	0
Chung-Reynolds	$f_2(x)=\left(\sum_{i=1}^{D}x_i^2\right)^2$	[-10, 10]	[-1, 1]	0
Powell-Sum	$f_3(x)=\sum_{i=1}^{D}\lvert x_i\rvert^{i+1}$	[-10, 10]	[-1, 1]	0
Rosenbrock	$f_4(x)=\sum_{i=1}^{D-1}\left(100\left(x_{i+1}-x_i^2\right)^2+\left(x_i-1\right)^2\right)$	[-10, 10]	[-1, 1]	0
Rotated Hyper-Ellipsoid	$f_5(x)=\sum_{i=1}^{D}\sum_{j=1}^{i}x_j^2$	[-10, 10]	[-1, 1]	0
Sphere	$f_6(x)=\sum_{i=1}^{D}x_i^2$	[-10, 10]	[-1, 1]	0
A Sum-Squares	$f_7(x)=\sum_{i=1}^{D}ix_i^2$	[-10, 10]	[-1, 1]	0
Zakharov	$f_8(x)=\sum_{i=1}^{D}x_i^2+\left(\sum_{i=1}^{D}0.5ix_i\right)^2+\left(\sum_{i=1}^{D}0.5ix_i\right)^4$	[-10, 10]	[-1, 1]	0

IV. FUNCTION EXPERIMENTS

A. EXPERIMENT PARAMETERS AND ENVIRONMENTS

The benchmark function experiment is a common method to measure the algorithm performance. To reflect the performance of ENBAS more accurately and comprehensively, this chapter selects eight benchmark functions that are widely used in mathematics fields to test the proposed algorithm. Therefore, the experimental results obtained by different benchmark functions can reflect the optimization ability of the proposed algorithm more objectively and comprehensively. These functions are divided into low dimension functions and high dimension functions, and all details of these functions are given in Table 1. In Table 1, D is the searching dimension and *Aim* is the ideal searching value. Generally, there is only one optimal solution in the low dimension function. The high dimension function has many locally optimal solutions because of the uneven distribution of local extreme points, the strong oscillation, and non-convexity, which enhances the problem-solving deception. In the original BAS literature, the author does not compare BAS with other algorithms. So to verify the ENBAS performance, this paper compares ENBAS with the bat algorithm (BA) [58], the cuckoo search algorithm (CS) [59], the simulated annealing algorithm (SA) [60], and the original BAS. BA, CS, and SA are the most classical optimization algorithms and are the hot-pot researches in engineering fields. They all use the iteration searching mechanism to find the optimal solution. All algorithm parameters were selected from the original algorithm literatures.

BA is a swarm intelligence optimization algorithm proposed by Xin-She Yang in 2010. The algorithm searches for the best solution by simulating the bat echolocation behavior. Because bats constantly adjust the searching frequency in the searching process, it speeds up the algorithm convergence speed. At the same time, bats adjust the pulse frequency and the echo loudness frequency of sound waves, which expands the exploration ability. For BA, the loudness coefficient was equal to 9, the rate coefficient was equal to 0. 9.

CS was proposed by Xin-She Yang. Cuckoos breed their eggs in other bird nests and use the mechanism of removing host eggs to increase the probability of their eggs hatched by the host. CS has advantages of a simple code, fewer parameters, easy to control and so on. In this chapter, the CS discovery probability was set as 0.25, and the step was set as 0.25.

SA comes from the physical annealing process in the metal, which is a local searching algorithm proposed in the early 1980s. SA is to heat the solid metal to a large temperature, so that atoms in the metal will be in a random state, and then slowly cool according to specific conditions. SA has two initial parameters, including initial temperature and attenuation factor. In this chapter, the parameter T_0 was equal to 100 and the parameter k was equal to 0.95. For ENBAS, the number of neighbors was three.

The population size and the maximum number of iterations were 20 and 1000, respectively. Record the best value, the worst value, the average value, and the variance. To obtain a fair comparison result and eliminate the randomness influence, each algorithm independently ran in MATLAB (R2014b) 10 times. The experimental environment was the Windows 7 operating system, Intel (R) Core (TM) i5-4210u CPU, 4GB. All programs, data, and charts were completed in MATLAB (R2014b).

B. NUMERICAL CALCULATION DISCUSSIONS

To verify the optimization effect of different algorithms, four indexes were selected to comprehensively evaluate searching results. Four indexes include *Best*, *Worst*, *Mean*, and *Var*. *Best* and *Worst* represent the best value and the worst value obtained by the algorithm in 10 independent runs. *Mean* represents the average value after 10 independent runs. When a group of data changed significantly, the average value

TABLE 2. Comparison of results for two dimension functions.

Function	Index	Algorithm				
		ENBAS	BAS	BA	CS	SA
$f_{1(D=2)}$	Best	0	7.8518E-05	1.5374E-05	0.0023	0.0003
	Worst	3.7903E-07	0.0005	0.0053	0.0283	0.0103
	Mean	6.8719E-08	0.0003	0.0010	0.0084	0.0036
	Var	1.6694E-14	1.7761E-08	2.67E-06	5.8446E-05	1.0434E-05
$f_{2(D=2)}$	Best	0	3.9231E-75	3.9603E-12	1.0686E-08	7.6242E-12
	Worst	0	3.2385E-72	2.6434E-09	4.4562E-05	3.1860E-08
	Mean	0	7.1337E-73	4.6663E-10	1.4011E-05	6.3354E-09
	Var	0	1.2406E-144	6.9849E-19	2.8174E-10	9.5460E-17
$f_{3(D=2)}$	Best	0	3.4840E-44	0.0001	0.0003	2.9952E-08
	Worst	0	1.3287E-38	0.0233	0.0149	5.2905E-06
	Mean	0	2.6477E-39	0.0115	0.0028	2.8922E-06
	Var	0	2.3228E-77	7.5033E-05	1.9773E-05	3.4786E-12
$f_{4(D=2)}$	Best	1.4585E-06	0.0018	1.1389E-06	0.0151	4.7478E-06
	Worst	6.0415E-05	0.0558	0.0008	0.6683	0.0030
	Mean	2.4866E-05	0.0116	0.0002	0.1352	0.0011
	Var	3.4054E-10	0.0003	6.1231E-08	0.0382	8.4141E-07
$f_{5(D=2)}$	Best	0	6.6624E-38	1.1496E-06	0.0044	1.6224E-05
	Worst	0	6.0181E-36	0.0003	0.0466	0.0006
	Mean	0	2.1591E-36	6.3345E-05	0.0172	0.0001
	Var	0	4.7982E-72	8.5699E-09	0.0002	2.8867E-08
$f_{6(D=2)}$	Best	0	1.8852E-37	0.0001	0.0003	1.9476E-05
	Worst	0	4.5317E-36	0.0182	0.0207	0.0003
	Mean	0	1.0965E-36	0.0075	0.0086	7.9192E-05
	Var	0	1.8247E-72	3.8074E-05	5.7417E-05	7.0543E-09
$f_{7(D=2)}$	Best	0	6.1182E-37	4.6370E-05	9.7127E-05	3.1388E-06
	Worst	0	3.6233E-36	0.1163	0.0337	0.0003
	Mean	0	1.2670E-36	0.0199	0.0124	8.4316E-05
	Var	0	7.8469E-73	0.0012	0.0001	1.1053E-08
$f_{8(D=2)}$	Best	0	9.2494E-37	5.2051E-07	0.0011	7.4554E-08
	Worst	0	5.2674E-25	9.1834E-05	0.0423	0.0002
	Mean	0	5.2674E-26	3.2775E-05	0.0206	7.4271E-05
	Var	0	2.7746E-50	9.3824E-10	0.0002	5.6280E-09

can be used to explain the centralized trend of the data. *Var* is the variance that is used to measure the deviation between a random variable and its mathematical expectation. In other words, the variance is the square value average of the difference between each sample value and the average of whole sample values. Variance can reflect the data dispersion degree and the robustness. The statistical results of algorithms on benchmark functions are shown in Table 2 to Table 4. It can be seen from the results that ENBAS can get the minimum value in different testing functions. Compared with BA, CS, and SA, the optimal value of each function is closer to *Aim*. For two dimension function, ENBAS can reach the ideal values in benchmark functions $f_{1(D=2)}$, $f_{2(D=2)}$, $f_{3(D=2)}$, $f_{5(D=2)}$, $f_{6(D=2)}$, $f_{7(D=2)}$, $f_{8(D=2)}$. For 50 dimension functions, ENBAS can reach the ideal values in benchmark functions $f_{1(D=50)}$, $f_{2(D=50)}$, $f_{3(D=50)}$, $f_{5(D=50)}$, $f_{6(D=50)}$, $f_{7(D=50)}$. For 100 dimensional function, ENBAS can reach the ideal value in benchmark functions $f_{1(D=100)}$, $f_{2(D=100)}$, $f_{3(D=100)}$, $f_{5(D=100)}$, $f_{6(D=100)}$, $f_{7(D=100)}$. In the function calculation results, the four indexes calculated by ENBAS are better than those of other algorithms. Function calculation results fully demonstrate the stability and the robustness of ENBAS, and show that the proposed algorithm in this paper is effective in solving low dimension and high dimension space problems.

C. THE WILCOXON RANK SUM TEST DISCUSSIONS

The rank sum test is a non-parametric statistical test used to define whether the results are statistically significant. The non-parametric statistical test is to use some parameters to describe the population characteristic and make some hypothesis tests on population properties. Compared with the parametric test, the non-parametric statistical test has no special requirements on the distribution of a group data and is often used to test the algorithm performance. The rank sum test arranges all the data in order from small to large. Because there is no special form of discrete data or known distribution, it has strong practicability. However, the rank sum test ignores the absolute value difference in the data testing, which not only makes the testing results approximate, but also causes the test information lossing. Wilcoxon improves the basic rank sum test by considering different directions and data sizes to check data differences and provides more effective performances than the basic rank sum test. The calculated result of the Wilcoxon rank sum test

TABLE 3. Comparison of results for 50 dimension functions.

Function	Index	Algorithm				
		ENBAS	BAS	BA	CS	SA
$f_1(D=50)$	Best	0	5.6723	5.1735	4.8807	9.7465
	Worst	1.6471E-06	7.1884	9.1255	7.7577	11.8840
	Mean	1.6471E-07	6.5436	7.9620	6.3742	10.6142
	Var	2.7128E-13	0.2507	1.4494	0.8708	0.4706
$f_2(D=50)$	Best	0	51.1839	35.0345	37.5886	100.7043
	Worst	0	71.4903	93.7240	74.1733	167.5735
	Mean	0	59.7830	75.2814	53.2535	138.7818
	Var	0	53.7455	275.6716	141.1763	427.1463
$f_3(D=50)$	Best	0	0.0326	0.1341	0.0898	0.4325
	Worst	0	0.3407	0.3727	0.6002	1.7222
	Mean	0	0.1928	0.2330	0.3787	1.1552
	Var	0	0.0122	0.0053	0.0297	0.1529
$f_4(D=50)$	Best	47.1830	101.2070	86.4847	808.5822	1.4884E+03
	Worst	48.9111	627.2158	1.1741E+03	1.1290E+03	2.0213E+03
	Mean	48.5601	377.7209	726.4236	951.6826	1.6795E+03
	Var	0.3029	3.9471E+04	1.0053E+05	1.2808E+04	2.6604E+04
$f_5(D=50)$	Best	0	122.9626	173.8676	135.5256	254.4603
	Worst	0	181.4231	231.2069	196.5923	306.5769
	Mean	0	154.0269	198.8674	170.1749	278.5001
	Var	0	363.4695	363.1673	395.0964	338.3090
$f_6(D=50)$	Best	0	7.1506	5.5485	6.0786	11.2563
	Worst	0	8.7624	9.9248	8.1187	14.6343
	Mean	0	7.8789	8.1876	7.2071	12.5894
	Var	0	0.2404	2.7082	0.6356	1.0176
$f_7(D=50)$	Best	0	125.3552	99.1539	103.2561	247.5866
	Worst	0	175.5646	218.1110	210.6979	315.7226
	Mean	0	156.4094	185.6848	174.4733	272.8647
	Var	0	288.5899	1.8026E+03	823.4397	396.8162
$f_8(D=50)$	Best	1.0393E-80	11.2808	10.9910	9.1762	13.1881
	Worst	9.6897E-61	14.4446	13.0726	13.6659	22.9603
	Mean	9.9839E-62	11.9685	12.1128	12.0346	20.0196
	Var	9.3341E-122	0.8114	0.6787	1.7647	7.7032

is called *P*-value. If *P*-value is less than 0.05, it is showed that there is a significant difference between two data groups at the level of 0.05. Due to the randomness of the swarm intelligence algorithm, statistical experiments must be carried out to ensure data validity [61], [62]. To further compare the performance of the proposed algorithms in this paper, the Wilcoxon rank sum test was computed. Table 5 shows the Wilcoxon rank sum test results. For ENBAS, all *P* values are less than or equal to 0.05, which further verifies the superior performance of ENBAS. The Wilcoxon rank sum test results show that the proposed algorithm has a strong searching efficiency and the maximum searching mechanism around the best solution, which further shows that the proposed algorithm has a good searching performance.

D. ITERATION DISCUSSIONS

The iteration is a series of feedback running processes, the iteration aim is to approach the desired result, and the result calculated by the algorithm after each iteration will be the initial value of the next iteration. This paper gave the average iteration curve of all algorithms after 10 independent operations as shown in Figure 3 to Figure 5. From iteration results, it can be concluded that ENBAS can achieve a certain optimization accuracy for most testing functions, which shows that the proposed algorithm has a strong global searching ability. Compared with other algorithms, the proposed algorithm has the fastest iteration speed in all iteration curves, which shows that ENBAS can approach the optimal solution more quickly in a shorter time. Because there are many local extremum points in high dimension functions, an algorithm often needs to traverse the whole searching space, which will cause the searching explosion phenomenon due to the too large local searching space, which will lead to the failure of completing the searching task within the specified time. For the high dimension function, the basic BAS performance is reduced very quickly, and ENBAS can keep the good searching accuracy and stability. From the high dimension iteration graph, it can be seen that the ENBAS iteration curve can converge to the optimal value at the iteration beginning. With the increasing of iteration times, comparison algorithms quickly trap in the local optimization area and the searching stagnation. Because of the increasing of population diversities, ENBAS can keep the optimal searching performance. Algorithm iteration curves show that

TABLE 4. Comparison of results for 100 dimension functions.

Function	Index	Algorithm				
		ENBAS	BAS	BA	CS	SA
$f_{1(D=100)}$	Best	0	15.9481	13.1212	11.4307	22.0760
	Worst	3.4183E-06	19.8402	21.0406	15.9354	27.9650
	Mean	3.4183E-07	17.6355	18.8812	14.4864	25.2634
	Var	1.1685E-12	1.1325	6.4548	1.4599	2.3899
$f_{2(D=100)}$	Best	0	331.9778	38.2643	141.7407	639.6525
	Worst	0	495.7652	466.1429	369.8894	837.3831
	Mean	0	422.2185	189.9797	253.3648	729.2288
	Var	0	3.3904E+03	2.3466E+04	4.9048E+03	3.6121E+03
$f_{3(D=100)}$	Best	0	0.0351	0.1569	0.2182	0.6636
	Worst	0	0.7013	0.6299	1.2769	1.6384
	Mean	0	0.3751	0.4295	0.4884	1.2357
	Var	0	0.0424	0.0323	0.0978	0.0833
$f_{4(D=100)}$	Best	98.0117	1.2197E+03	418.5633	1.5702E+03	2.0750E+03
	Worst	98.8811	1.7342E+03	2.3842E+03	2.5042E+03	2.7107E+03
	Mean	98.5559	1.4724E+03	1.6134E+03	2.1807E+03	2.4133E+03
	Var	0.1395	3.5245E+04	4.9026E+05	8.6330E+04	3.9937E+04
$f_{5(D=100)}$	Best	0	803.0874	48.9441	497.0659	1.2515 E+03
	Worst	0	1.0452E+03	946.8220	865.0815	1.4940 E+03
	Mean	0	922.1134	597.5373	733.0056	1.4145 E+03
	Var	0	4.4908E+03	8.1471E+04	1.3668E+04	4.9401 E+03
$f_{6(D=100)}$	Best	0	16.3335	11.1344	14.2491	19.5668
	Worst	0	23.0020	22.3598	17.3137	24.4237
	Mean	0	19.9721	18.1428	15.9821	21.9981
	Std	0	2.8092	11.5830	1.0404	2.0815
$f_{7(D=100)}$	Best	0	830.7103	4.1381	537.7166	1.2613E+03
	Worst	0	1.1052E+03	659.0103	870.0702	1.4751E+03
	Mean	0	939.3762	412.0031	710.0642	1.3540E+03
	Var	0	8.3490E+03	4.3366E+04	1.1920E+04	4.6271E+03
$f_{8(D=100)}$	Best	1.6763E-24	25.1870	26.0671	18.1842	14.7611
	Worst	2.9906E-16	27.8549	29.1124	34.5088	21.8563
	Mean	4.5109E-17	26.1739	27.6508	27.6104	17.9833
	Var	9.2784E-33	0.9637	1.0996	21.9927	5.5560

ENBAS has a strong ability to explore the feasible region and escape the local solution in the searching range, and can avoid the searching extreme point effectively.

E. BOX PLOT DISCUSSIONS

The box plot is used to show a set of discrete data. In the algorithm analysis, it is mainly used to reflect the symmetry and distribution of data. The typical box plot has six parameters including the maximum value, the minimum value, the median value, the upper quartile, the lower quartile, and the abnormal value. A set of data can be evaluated by these parameters. The upper and lower boxes in the box plot are the upper quartile and the lower quartile, and the middle horizontal line of the box is the data median. Two horizontal lines in the box plot are the upper and lower edges of a group data, and the discrete points in the figure are data abnormal values. Figure 6 to Figure 8 show all box plots of different algorithms after 10 independent operations. Because there are a lot of local solutions in high dimension functions, so the solution aggregation degree is an important index to evaluate the algorithm performance. All ENBAS box plots are narrower than those of BAS. The box plot results show that the ENBAS not only has a good searching accuracy, an excellent robustness, and the stability, but also can avoid the local extremum in high dimension functions. In the single peak function or the low dimension function with a long peak distance, ENBAS also can obtain the optimal individual information to enhance the searching effectiveness. In the multiple local peak high-dimension optimization functions, the elite selection mechanism in ENBAS can search for a better position and get a new searching space by selecting the optimal location information in the population.

F. SEARCH PATH DISCUSSIONS

The searching path can test whether the algorithm falls into the local solution areas. To further prove the powerful

TABLE 5. Comparison of the wilcoxon rank sum test.

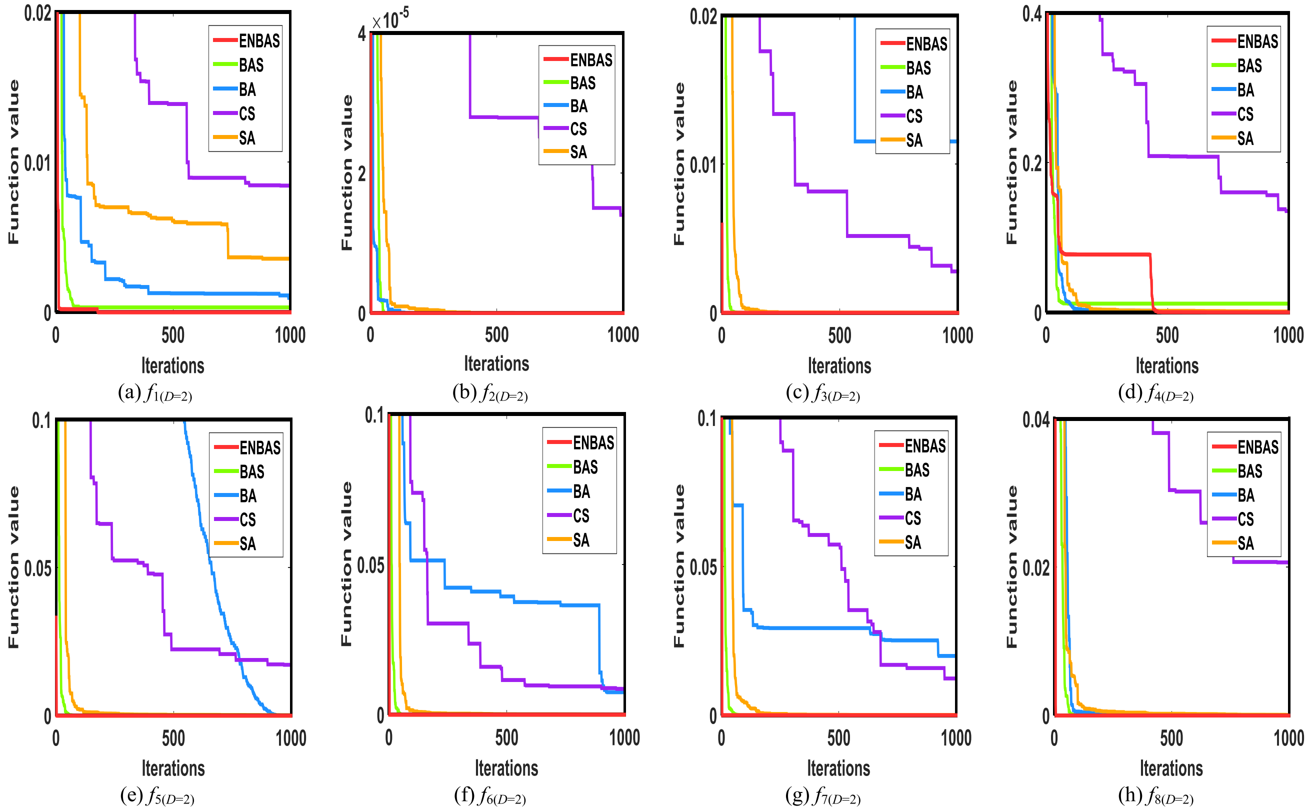
Function	Algorithm			
	BAS	BA	CS	SA
f_1	1.32 E-04	1.32 E-04	1.32 E-04	1.32 E-04
f_2	6.39E-05	6.39E-05	6.39E-05	6.39E-05
f_3	6.39E-05	6.39E-05	6.39E-05	6.39E-05
f_4	1.83E-04	0.05	1.83E-04	0.01
f_5	6.39E-05	6.39E-05	6.39E-05	6.39E-05
f_6	6.39E-05	6.39E-05	6.39E-05	6.39E-05
f_7	6.39E-05	6.39E-05	6.39E-05	6.39E-05
f_8	6.39E-05	6.39E-05	6.39E-05	6.39E-05
$f_{1(D=50)}$	8.74E-05	8.74E-05	8.74E-05	8.74E-05
$f_{2(D=50)}$	6.39E-05	6.39E-05	6.39E-05	6.39E-05
$f_3\ (D=50)$	6.39E-05	6.39E-05	6.39E-05	6.39E-05
$f_4\ (D=50)$	1.83E-04	1.83E-04	1.83E-04	1.83E-04
$f_5\ (D=50)$	6.39E-05	6.39E-05	6.39E-05	6.39E-05
$f_6\ (D=50)$	6.39E-05	6.39E-05	6.39E-05	6.39E-05
$f_7\ (D=50)$	6.39E-05	6.39E-05	6.39E-05	6.39E-05
$f_8\ (D=50)$	1.83E-04	1.83E-04	1.83E-04	1.83E-04
$f_{1(D=100)}$	8.74E-05	8.74E-05	8.74E-05	8.74E-05
$f_{2(D=100)}$	6.39E-05	6.39E-05	6.39E-05	6.39E-05
$f_3\ (D=100)$	6.39E-05	6.39E-05	6.39E-05	6.39E-05
$f_4\ (D=100)$	1.83E-04	1.83E-04	1.83E-04	1.83E-04
$f_5\ (D=100)$	6.39E-05	6.39E-05	6.39E-05	6.39E-05
$f_6\ (D=100)$	6.39E-05	6.39E-05	6.39E-05	6.39E-05
$f_7\ (D=100)$	6.39E-05	6.39E-05	6.39E-05	6.39E-05
$f_8\ (D=100)$	1.83E-04	1.83E-04	1.83E-04	1.83E-04

searching ability of the proposed algorithm in this chapter, the optimal ENBAS searching path and the optimal BAS searching path refracted to the two dimension plane and the contour map on the two dimension plane is shown in Figure 9. The yellow spot in the figure is the theoretical optimal position. It can be seen from Figure 9 that the optimal ENBAS searching path is much smaller than the BAS searching path, and there is a large number of short-repeat paths and occasional invalid-long paths in the BAS searching path. Searching paths show that the ENBAS searching range compared with the basic BAS is wider and less affected by the number of iterations. Therefore, ENBAS has a better optimization ability and can consider the searching speed and accuracy. And ENBAS can not only provide a satisfactory solution to the problem as far as possible under the condition of meeting variable demand but also solve the extremum of nonlinear multimodal function more quickly. The proposed algorithm can save a lot of computing time, and only need a few iterations to get a better path without too much calculation, which also shows that ENBAS has a high positioning accuracy, can strengthen the early searching ability of the original algorithm, prevent excessive attenuation and the premature phenomenon.

V. CEC TEST SUITS EXPERIMENTS

A. EXPERIMENT PARAMETERS AND ENVIRONMENTS

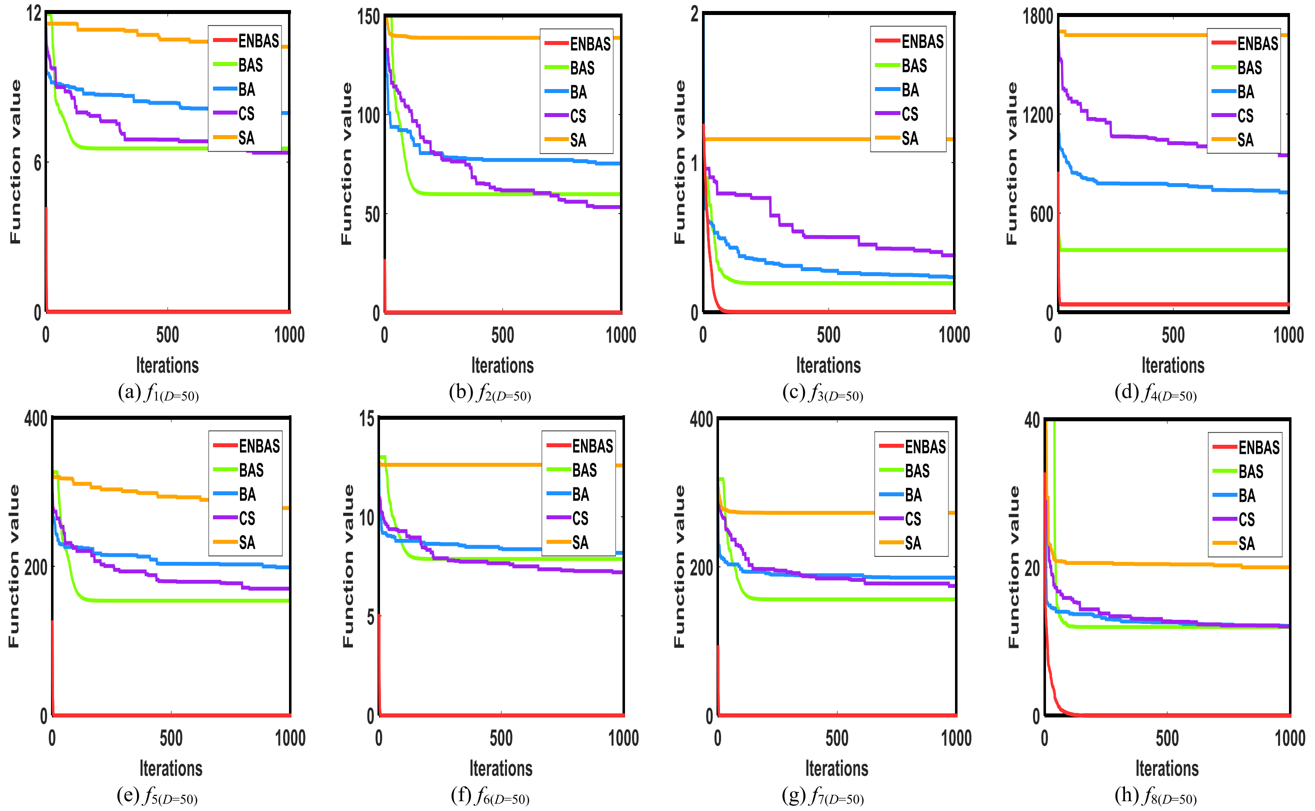
To show the proposed algorithm testing performance, this paper compared ENBAS with recent optimization algorithms that were proposed in the last 10 years and surely highly-cited ones. Testing functions selected testing suits in 2018 IEEE congress on Evolutionary Computation (CEC) conference. CEC testing suits are widely attracted many researchers for testing their developed algorithms [63]. In CEC testing suits, there are unimodal functions and multimodal functions. Unimodal functions include Bent Cigar function and Zakharov Function. Multimodal functions include Rosenbrock Function, Rastrigin Function, Expanded Scaffer's F6 Function, Levy Function, and Schwefel's Function. To further show the proposed algorithm comprehensively, this paper selected basic unimodal testing functions in f_9 to f_{15}. All function dimensions were 30, and the scope selected in the range of $[-1, 1]$. Compared algorithms in this paper selected butterfly optimization algorithm (BOA) [64], grey wolf optimizer (GWO) [65], the lévy-flight salp swarm algorithm (LSSA) [66], sine cosine algorithm (SCA) [67], salp swarm algorithm (SSA) [68], improved SSA based on weight factor and adaptive mutation (WASSA) [69], whale



(a) $f_{1(D=2)}$ (b) $f_{2(D=2)}$ (c) $f_{3(D=2)}$ (d) $f_{4(D=2)}$

(e) $f_{5(D=2)}$ (f) $f_{6(D=2)}$ (g) $f_{7(D=2)}$ (h) $f_{8(D=2)}$

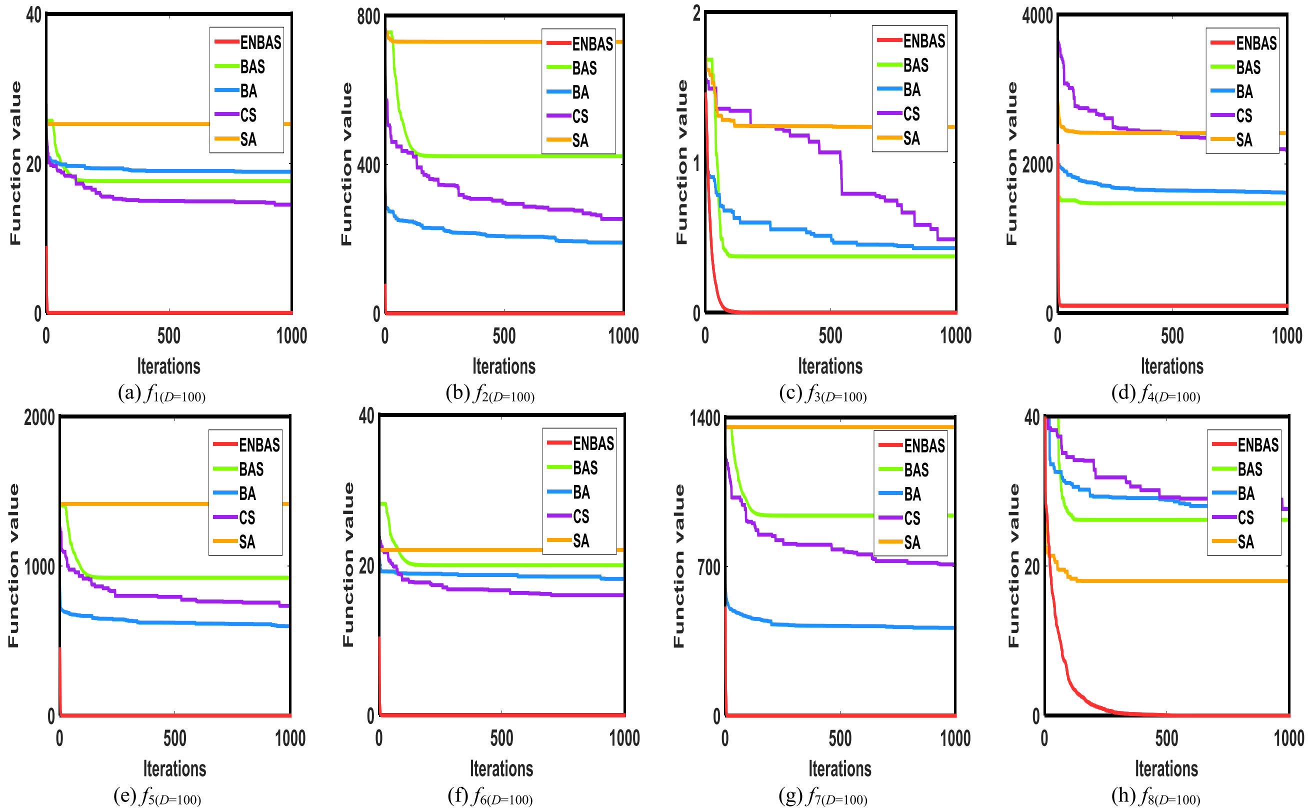
FIGURE 3. Average convergence curves of 2 dimension functions.



(a) $f_{1(D=50)}$ (b) $f_{2(D=50)}$ (c) $f_{3(D=50)}$ (d) $f_{4(D=50)}$

(e) $f_{5(D=50)}$ (f) $f_{6(D=50)}$ (g) $f_{7(D=50)}$ (h) $f_{8(D=50)}$

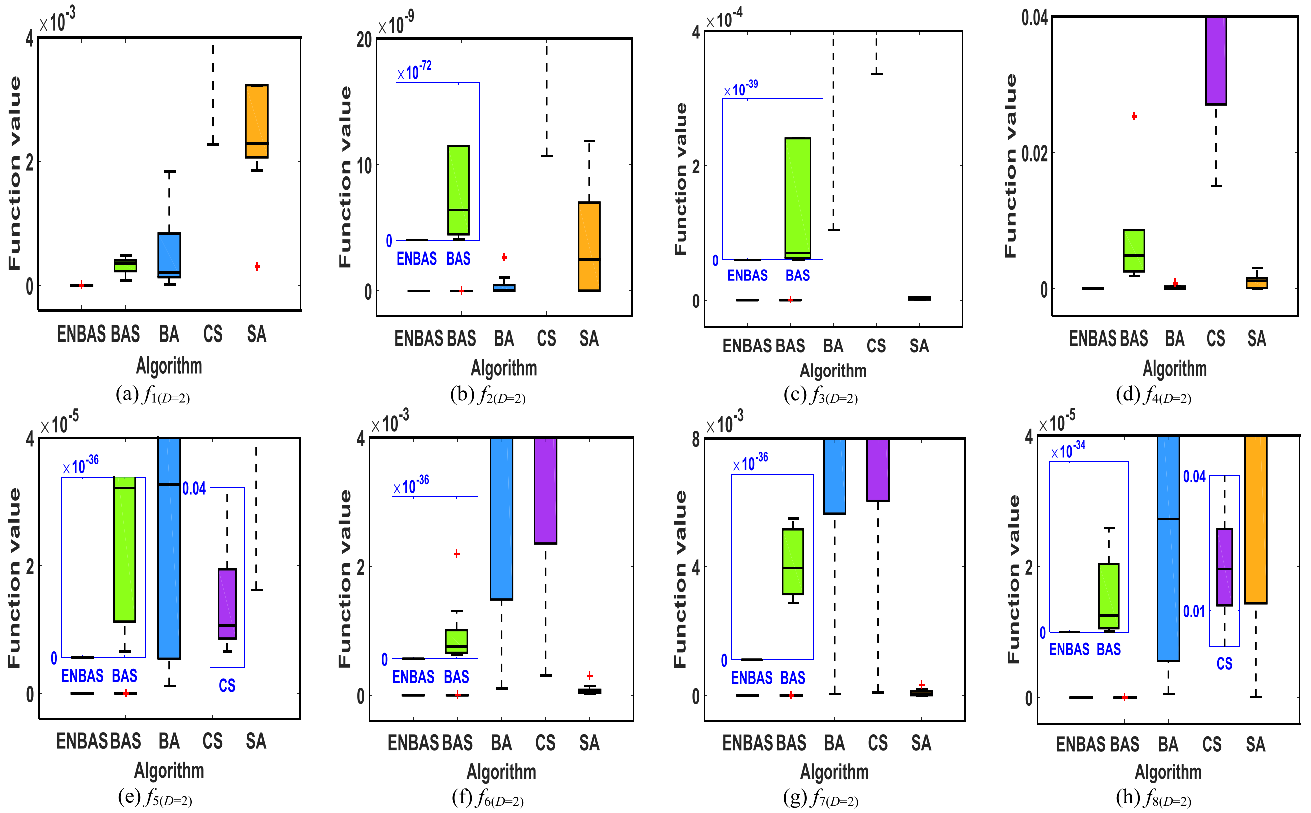
FIGURE 4. Average convergence curves of 50 dimension functions.



(a) $f_{1(D=100)}$ (b) $f_{2(D=100)}$ (c) $f_{3(D=100)}$ (d) $f_{4(D=100)}$

(e) $f_{5(D=100)}$ (f) $f_{6(D=100)}$ (g) $f_{7(D=100)}$ (h) $f_{8(D=100)}$

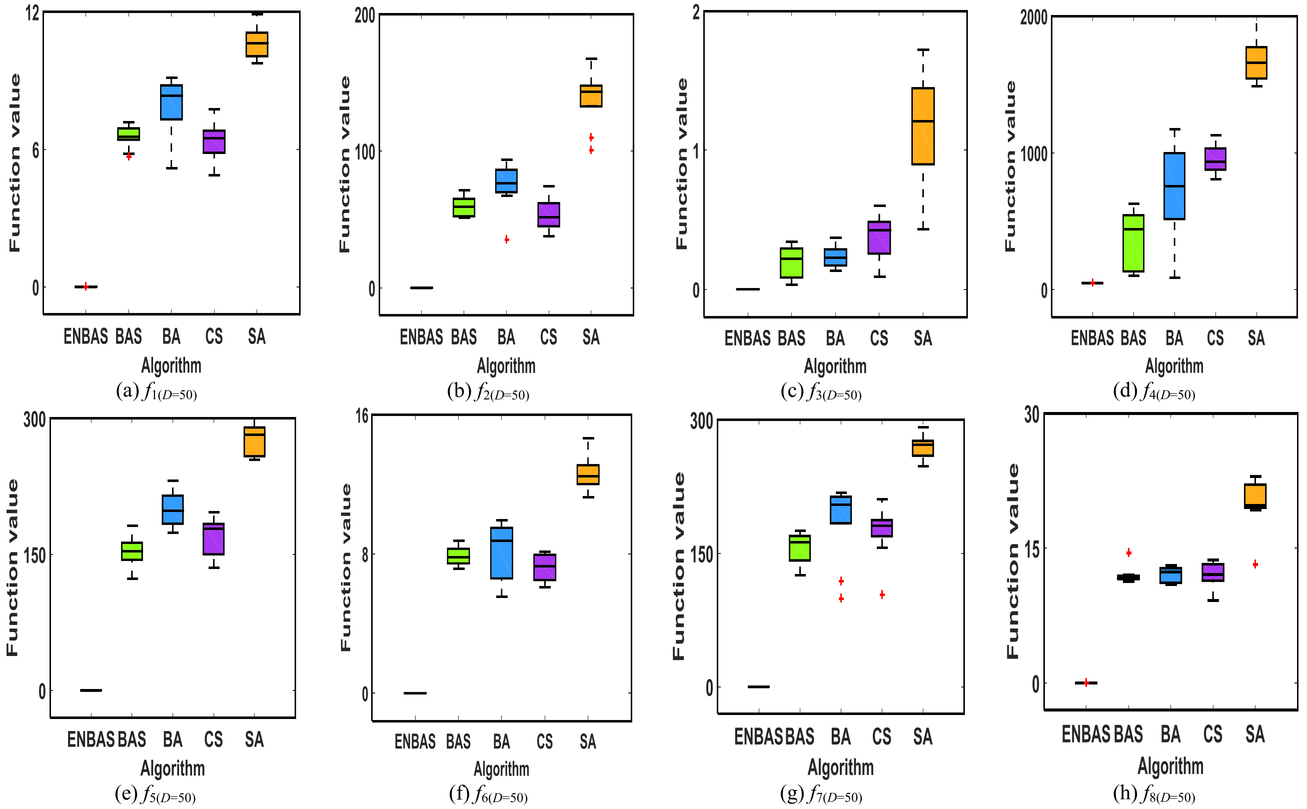
FIGURE 5. Average convergence curves of 100 dimension functions.



(a) $f_{1(D=2)}$ (b) $f_{2(D=2)}$ (c) $f_{3(D=2)}$ (d) $f_{4(D=2)}$

(e) $f_{5(D=2)}$ (f) $f_{6(D=2)}$ (g) $f_{7(D=2)}$ (h) $f_{8(D=2)}$

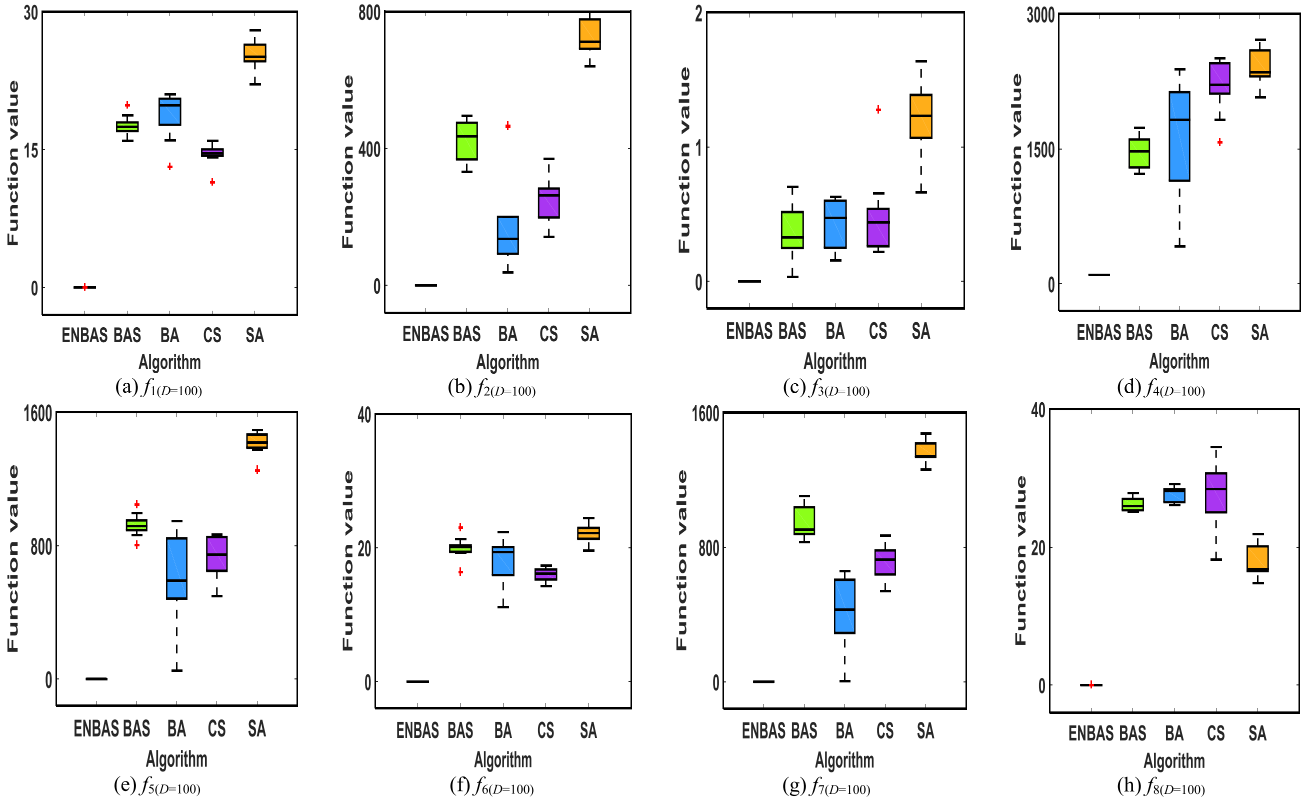
FIGURE 6. Box-plot charts of 2 dimension functions.



(a) $f_{1(D=50)}$ (b) $f_{2(D=50)}$ (c) $f_{3(D=50)}$ (d) $f_{4(D=50)}$

(e) $f_{5(D=50)}$ (f) $f_{6(D=50)}$ (g) $f_{7(D=50)}$ (h) $f_{8(D=50)}$

FIGURE 7. Box-plot charts of 50 dimension functions.



(a) $f_{1(D=100)}$ (b) $f_{2(D=100)}$ (c) $f_{3(D=100)}$ (d) $f_{4(D=100)}$

(e) $f_{5(D=100)}$ (f) $f_{6(D=100)}$ (g) $f_{7(D=100)}$ (h) $f_{8(D=100)}$

FIGURE 8. Box-plot charts of 100 dimension functions.

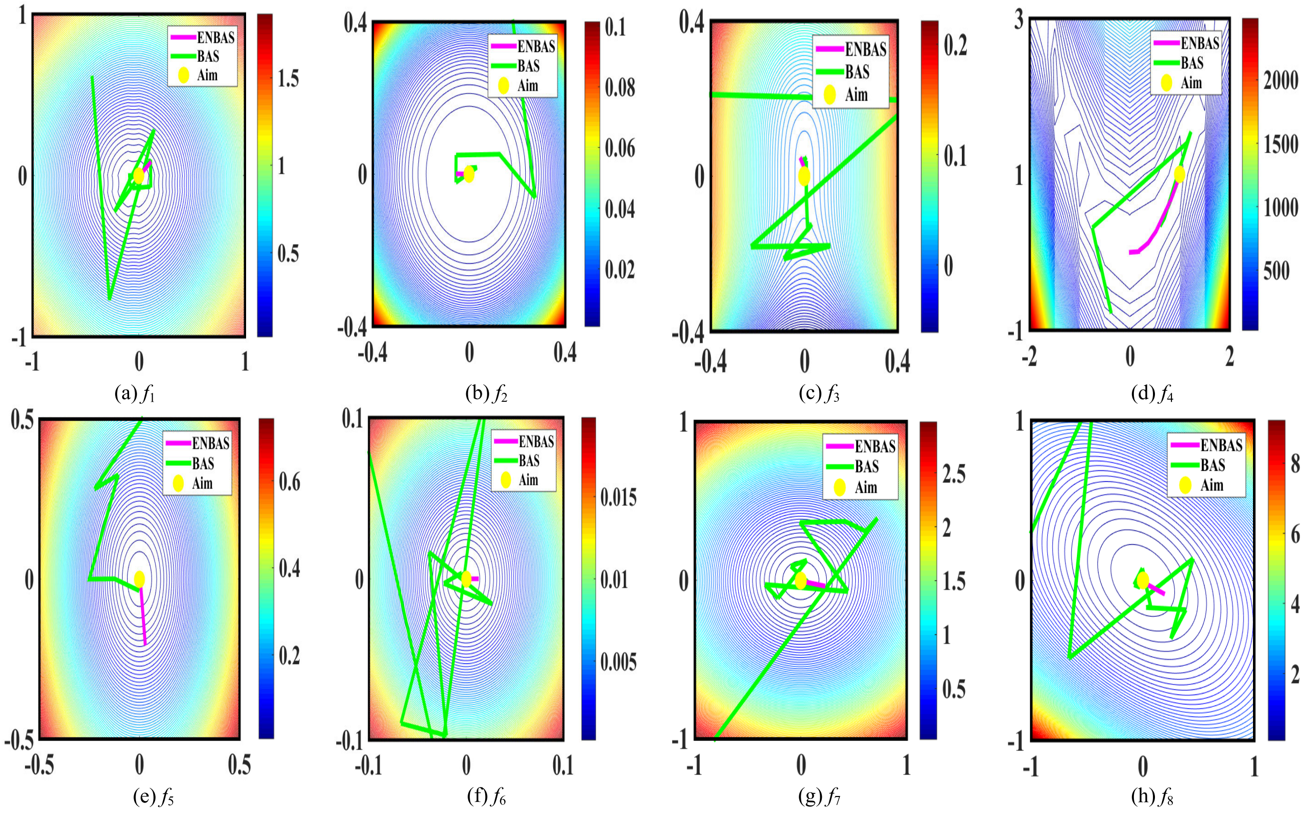


FIGURE 9. **Algorithm searching paths.**

optimization algorithm (WOA) [70]. BOA was proposed by Arora and Singh, modular modality c selected 0.01 and power exponent a selected from 0.1 to 0.3. The switch probability p selected 0.8. GWO inspired by grey wolve living habits was proposed in 2014. For GWO, r_1 and r_2 selects in the range of [0, 1]. LSSA was proposed by Zhikai Xing and Heming Jia in 2019. The power-law exponent β selected 1.5. P selected 0.5. SCA was proposed by Seyedali Mirjalili in 2016. For SCA, $a = 2$, $r_2 = 2\pi$, r_3 and r_4 selected in the range of [0, 1]. SSA was proposed by Seyedali Mirjalili in 2017. The parameter c_2 and c_3 were in the range of [0, 1]. Probability selected 0.5. WASSA was proposed by Jun Wu in 2019. Parameters w_{max} selected 1, w_{min} selected 0. WOA was proposed by Seyedali Mirjalili and Andrew Lewis in 2016. For WOA, r and p selected in the range of [0, 1], l selects in the range of [−1, 1], b selected 2. In this paper, all initial parameter values of all algorithms were selected according to original algorithm literature, and all algorithm procedures and details could be found in the original algorithm literatures. The population size and the maximum number of iterations were 20 and 1000, respectively. Each algorithm independently ran in MATLAB (R2014b) 10 times. The experimental environment was the Windows 7 operating system, Intel (R) Core (TM) i5-4210u CPU, 4GB. All programs, data, and charts were completed in MATLAB (R2014b). Testing results were showed in Table 6.

B. NUMERICAL CALCULATION DISCUSSIONS

Although ENBAS can effectively solve the extremum in some testing functions, it is not always able to find the theoretical optimal solution. And when the optimization functions are multi-dimensional functions, the solution accuracy is significantly reduced, and the optimal solution is unstable and the fluctuation range is large. For basic multimodal functions, ENBAS can reach the ideal optimal values in f_9, f_{12}, f_{13}, f_{15}. In f_{11}, ENBAS gives a poor result, and results of LSSA, SCA, SSA, WASSA, are better than those of ENBAS. In f_{14}, results of LSSA, SCA, SSA, WASSA, WOA are better than those of ENBAS. In general, the calculation results of the proposed algorithm in this paper is better than those of other algorithms, which fully shows the stability and the robustness of ENBAS, and displays that the proposed algorithm in this paper is effective in solving numerical calculation problems.

C. FRIEDMAN TEST AND NEMENYI TEST

A full statistical analysis of the comparison was presented based on significance non-parametric tests in this paper. This paper used the Friedman test and the Nemenyi test to compare algorithm performances. Friedman test, was proposed by M. Friedman in 1973, is a statistical test for the homogeneity of multiple samples, and is usually used to compare whether there are significant differences among different levels of multiple factors. Friedman test does not require samples

TABLE 6. Comparison of results for CEC functions.

Function	Index	Algorithm							
		ENBAS	BOA	GWO	LSSA	SCA	SSA	WASSA	WOA
f_9	Best	0	3.9490E-05	2.7519E-40	4.3436E-10	8.7847E-198	1.8434E-15	1.2017E-46	1.1503E-23
	Worst	0	5.3301E-05	7.0082E-25	5.3414E-07	1.9129E-122	1.1855E-08	1.3370E-37	3.3660E-12
	Mean	0	4.6132E-05	7.4776E-26	1.1338E-07	3.6292E-123	4.5968E-09	1.3697E-38	3.3746E-13
	Var	0	1.9937E-11	4.8584E-50	3.4270E-14	5.8752E-245	2.3854E-17	1.7784E-75	1.1323E-24
f_{10}	Best	2.4996E-172	0.0021	9.1047E-20	0.0013	1.8666E-122	3.5670E-05	1.1162E-25	1.6756E-06
	Worst	4.8822E-139	0.0027	3.9902E-09	0.1194	2.5323E-10	0.0275	3.6662E-24	1.4088E-04
	Mean	5.3807E-140	0.0023	5.9284E-10	0.0158	2.5323E-11	0.0056	8.8057E-25	3.7521E-05
	Var	2.3539E-278	3.9125E-08	1.6024E-18	0.0013	6.4125E-21	6.4353E-05	1.2851E-48	1.7406E-09
f_{11}	Best	27.1802	28.8931	28.9684	0	0	0	0	0
	Worst	28.9203	28.9746	29.0000	0	28.8740	0	0	131.0609
	Mean	28.4873	28.9461	28.9886	0	14.3928	0	0	36.8752
	Var	0.4826	0.0007	9.3681E-05	0	230.1733	0	0	1.5043E+03
f_{12}	Best	0	0	0	5.5894E-07	0	1.1369E-13	0	15.9665
	Worst	0	0	9.4055E-07	0.0018	0	0.0005	0	21.9525
	Mean	0	0	1.1860E-07	0.0005	0	0.0005	0	20.7379
	Var	0	0	8.9266E-14	4.5821E-07	0	3.3319E-08	0	3.4760
f_{13}	Best	0	0.0021	0	1.3513E-05	0	1.9405E-07	0	3.3307E-16
	Worst	0	0.0024	1.5469E-11	1.3820E-04	0	2.5744E-05	0	6.5392E-14
	Mean	0	0.0023	5.0243E-12	4.9044E-05	0	1.1756E-05	0	1.5554E-14
	Var	0	1.0596E-08	3.1913E-23	1.6815E-09	0	9.4974E-11	0	5.2282E-28
f_{14}	Best	1.2713	2.7280	2.3173	1.4998E-32	1.4998E-32	1.4998E-32	1.4998E-32	1.4998E-32
	Worst	2.1698	3.0655	3.1550	1.4998E-32	0.0909	1.4998E-32	1.4998E-32	1.4998E-32
	Mean	1.7545	2.9693	2.8372	1.4998E-32	0.0091	1.4998E-32	1.4998E-32	1.4998E-32
	Var	0.0922	0.0108	0.0784	0	8.2568E-04	0	0	0
f_{15}	Best	0.0004	0.0013	0.0004	0.0004	0.0004	0.0004	0.0004	0.0004
	Worst	0.0004	0.0014	0.0004	0.0004	0.0004	0.0004	0.0004	0.0004
	Mean	0.0004	0.0014	0.0004	0.0004	0.0004	0.0004	0.0004	0.0004
	Var	0	1.2996E-09	2.6470E-23	1.4251E-12	0	2.9039E-11	0	1.8529E-23

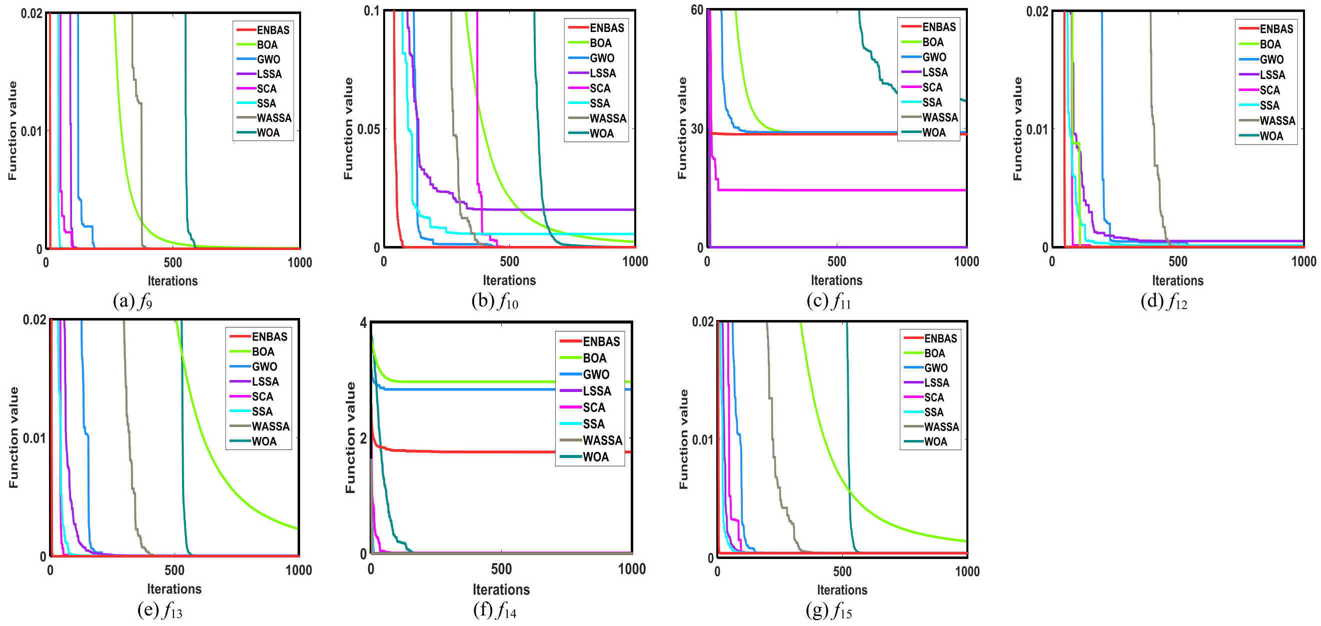
TABLE 7. Comparison of the Friedman test.

Index	Function						
	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}
p-value	0.004	0.005	0.017	0.005	0.005	0.005	0.004

TABLE 8. Comparison of the Nemenyi test.

Function	Algorithm						
	BOA	GWO	LSSA	SCA	SSA	WASSA	WOA
f_9	0.001	0.139	0.001	0.001	0.900	0.533	0.006
f_{10}	0.001	0.139	0.001	0.900	0.001	0.697	0.005
f_{11}	0.900	0.418	0.078	0.724	0.078	0.078	0.900
f_{12}	0.900	0.588	0.003	0.900	0.023	0.900	0.001
f_{13}	0.001	0.303	0.001	0.900	0.008	0.900	0.448
f_{14}	0.806	0.900	0.099	0.173	0.099	0.099	0.099
f_{15}	0.001	0.900	0.017	0.900	0.053	0.900	0.900

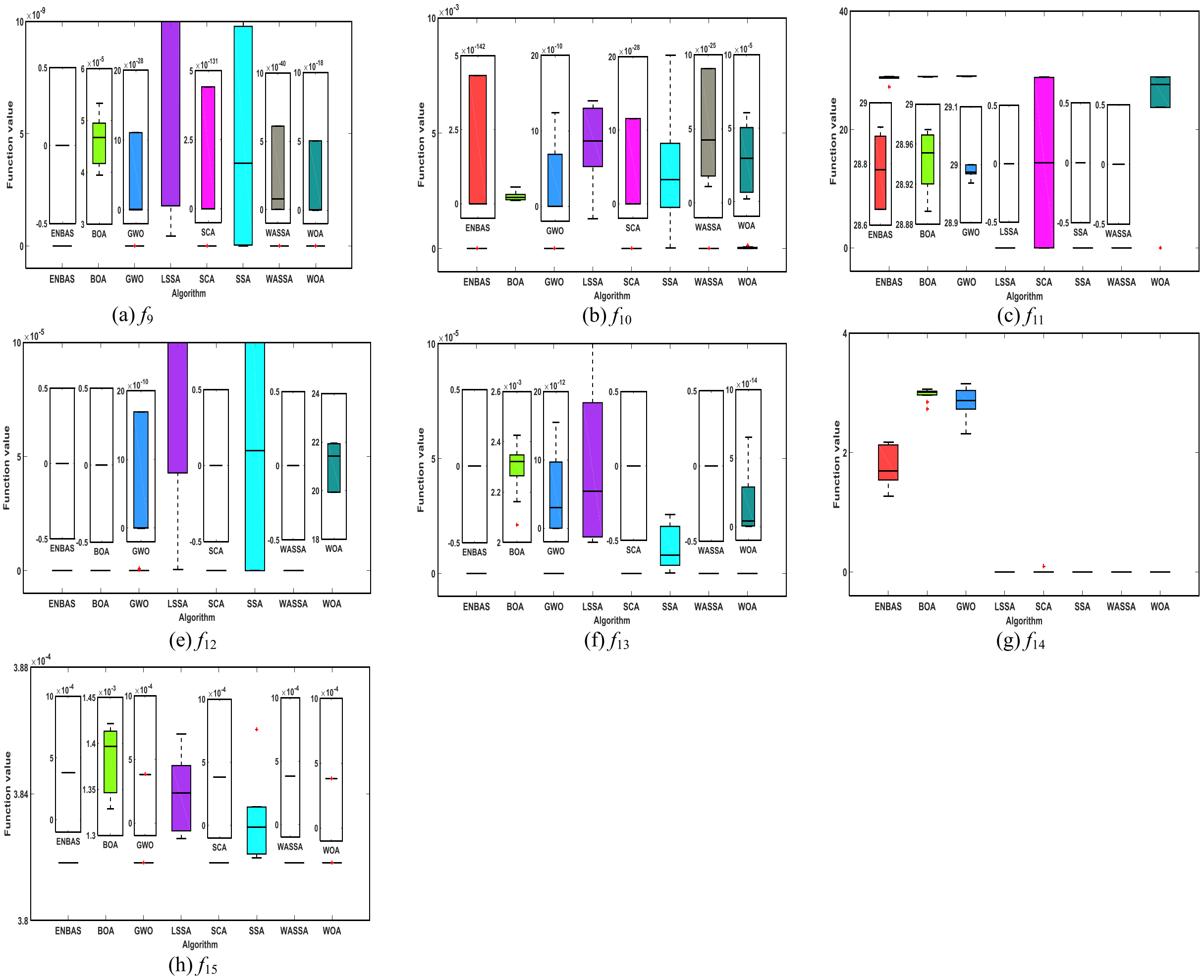
to obey the normal distribution, only uses the rank in different arrays, so it is necessary to convert the original data into the rank and calculate the average rank. Nemenyi test is a common method of the rank-sum test for multiple comparisons among multiple samples, and is suitable for comparing the performance of two algorithms. The calculated result of the Friedman test and Nemenyi test is called P-value. P-value means that there is a significant difference between two data groups at the level of P. Table 7 gives the Friedman test results in different algorithms. Table 8 gives the Nemenyi test results for different algorithms. From Table 7 we can find that all Friedman test results is less than 0.05, it can be



(a) f_9 (b) f_{10} (c) f_{11} (d) f_{12}

(e) f_{13} (f) f_{14} (g) f_{15}

FIGURE 10. Average convergence curves of 30 dimension functions.



(a) f_9 (b) f_{10} (c) f_{11}

(e) f_{12} (f) f_{13} (g) f_{14}

(h) f_{15}

FIGURE 11. Box-plot charts of 30 dimension functions.

TABLE 9. Comparison of results for HS-ES.

Algorithm	Function									
	f_{16}	f_{17}	f_{18}	f_{19}	f_{20}	f_{21}	f_{22}	f_{23}	f_{24}	f_{25}
HS-ES	2.04E+02	1.00E+02	3.06E+02	3.32E+02	4.43E+02	3.00E+02	4.77E+02	5.84E+02	2.74E+02	7.96E+02
ENBAS	2.12E+02	1.00E+02	3.13E+02	4.01E+02	4.43E+02	3.03E+02	4.52E+02	6.07E+02	3.09E+02	1.02E+03

seen that performances of different algorithms is obviously different. In other words, there is a significant difference at a level of 0.05. From Table 8 we can find that all Nemenyi test results of GWO and WASSA is larger than 0.05, which shows that the significant difference was small in GWO and WASSA under the Nemenyi test. Except for function f_9, Nemenyi test results of SCA is larger than 0.05. But all Nemenyi test result is less than 1, which shows that there are some significant differences, in other words, there are still some differences between the proposed algorithm and compared algorithms.

D. ITERATION DISCUSSIONS

This paper gives average iteration curves of each algorithm after 10 independent runs, as shown in Figure 10. For the functions $f_9, f_{10}, f_{12}, f_{13}, f_{15}$, the ENBAS iteration speed is obviously faster than other comparison algorithms. ENBAS can search the left and right sides of the optimal value through the initial large searching step, and can skip a certain range of obstacles in the process of tightening to the middle. The improved algorithm can keep the good convergence and optimization accuracy in the whole iteration process, mainly because ENBAS can enhance the randomness of the original algorithm, and can ensure the diversity of the population to make it jump out of the local optimum as far as possible. It can be seen that in the optimization process, the improved algorithm is always close to the optimal value quickly, and then the normal iteration is performed, which makes the proposed algorithm avoid the local premature in the optimization process.

E. BOX PLOT DISCUSSIONS

Figure 11 shows all box plots of different algorithms after 10 independent runs. ENBAS has the narrowest block diagram, the least outliers, the lowest median and upper and lower quartiles in functions $f_9, f_{10}, f_{12}, f_{13}, f_{15}$. The ENBAS box plot on functions $f_9, f_{12}, f_{13}, f_{15}$ is a straight line, that is to say that the algorithm has achieved the theoretical optimal value after 10 independent runs, and the optimal solution has a little influence, and experimental results have no significant difference.

F. RANKING DISCUSSIONS

In CEC 2018 competition, the winner was the hybrid sampling-evolution strategy (HS-ES) that combined the covariance matrix adaptation evolution strategy (CMA-ES) and the univariate sampling strategy, and HS-ES was proposed Geng Zhang and Yuhui Shi [71]. To further more show the proposed algorithm testing performance, this paper tested other CEC 2018 composition functions in f_{16} to f_{25} corresponding f_{21} to f_{30} in CEC 2018. And this paper also compared the proposed method with the winner of the CEC2018 competition. The population size and the maximum number of iterations were 50 and 20000, respectively. The scope was in the range of $[-50, 50]$. The searching dimension was equal to 10. The results of algorithms were shown in Table 9. It can be seen from Table 9 that ENBAS can get the equal results with HS-ES in f_{17} and f_{20}. ENBAS can get a better result than HS-ES in f_{22}. Other calculation results in ENBAS are worse than those of HS-ES. HS-ES is 2018 competition winner, it has the best performance in all competition algorithms. Although most testing results of HS-ES is better than the proposed method, there is no single algorithm that can solve all problems. Each algorithm has its advantages and disadvantages. Table 9 shows that the proposed algorithm has some competitive powers in some testing functions.

VI. CONCLUSION

Because the basic BAS does not consider the individual difference and the dynamic information in the searching process, this paper proposed a new algorithm based on the elite selection mechanism and the neighbor mobility strategy. Firstly, the Euclidean distance between the individual fitness value and the optimal individual fitness value is calculated to determine whether it is lower than the pre-determined threshold value. If the Euclidean distance is lower than the threshold, the individual position will be replaced to dynamically increase the population diversity, and elite individuals having the good convergence and the robustness will be selected in the whole population to guide other individuals to explore better positions. In the individual movement, the neighbor mobility strategy is used to exchange the information between neighboring individuals, which can make the individual deviate from the original moving orbit, and avoid the individual falling into the local area. Through different operations, the population diversity can be kept within a certain threshold, so that the proposed algorithm has a strong searching ability in each iteration.

The author thinks that the following aspects should be further discussed. The research on the discrete BAS algorithm is still few, but some optimization problems to be solved in the actual industry are discrete. The discrete optimization problem refers to that variables must be limited to integer in algorithm optimization. Most popular methods to solve discrete optimization problems are suitable for the integer-linear programming, and the discrete optimization problem has a great dependence on application fields. Therefore, it is very important to use the discrete BAS algorithm with a high searching precision to solve optimization problems in industrial fields.

REFERENCES

[1] I. Boussaïd, J. Lepagnot, and P. Siarry, ''A survey on optimization metaheuristics,'' *Inf. Sci.*, vol. 237, pp. 82–117, Jul. 2013.

[2] S. Arora, H. Singh, M. Sharma, S. Sharma, and P. Anand, ''A new hybrid algorithm based on Grey wolf optimization and crow search algorithm for unconstrained function optimization and feature selection,'' *IEEE Access*, vol. 7, pp. 26343–26361, 2019.

[3] A. Slowik and H. Kwasnicka, ''Nature inspired methods and their industry applications-swarm intelligence algorithms,'' *IEEE Trans. Ind. Informat.*, vol. 14, no. 3, pp. 1004–1015, Mar. 2018.

[4] X. Zhao, J. Zhang, and X. Qin, ''LOMA: A local outlier mining algorithm based on attribute relevance analysis,'' *Expert Syst. Appl.*, vol. 84, pp. 272–280, Oct. 2017.

[5] G.-G. Wang, A. H. Gandomi, A. H. Alavi, and S. Deb, ''A hybrid method based on krill herd and quantum-behaved particle swarm optimization,'' *Neural Comput. Appl.*, vol. 27, no. 4, pp. 989–1006, May 2016.

[6] X. Li, J. Wang, and M. Yin, ''Enhancing the performance of cuckoo search algorithm using orthogonal learning method,'' *Neural Comput. Appl.*, vol. 24, no. 6, pp. 1233–1247, May 2013.

[7] S. Zhang and S. Liu, ''A discrete improved artificial bee colony algorithm for 0–1 knapsack problem,'' *IEEE Access*, vol. 7, pp. 104982–104991, 2019.

[8] G. G. Wang, A. H. Gandomi, X. J. Zhao, and H. C. E. Chu, ''Hybridizing harmony search algorithm with cuckoo search for global numerical optimization,'' *Soft Comput.*, vol. 20, no. 1, pp. 273–285, Jan. 2016.

[9] M. Mavrovouniotis, C. Li, and S. Yang, ''A survey of swarm intelligence for dynamic optimization: Algorithms and applications,'' *Swarm Evol. Comput.*, vol. 33, pp. 1–17, Apr. 2017.

[10] G. Wang and Y. Tan, ''Improving metaheuristic algorithms with information feedback models,'' *IEEE Trans. Cybern.*, vol. 49, no. 2, pp. 542–555, Feb. 2019.

[11] L. Zuo, M. Yan, Y. Guo, and W. Ma, ''An improved KF-RBF based estimation algorithm for coverage control with unknown density function,'' *Complexity*, vol. 2019, Jul. 2019, Art. no. 6268127.

[12] F. A. Hashim, K. Hussain, E. H. Houssein, M. S. Mabrouk, and W. Al-Atabany, ''Archimedes optimization algorithm: A new metaheuristic algorithm for solving optimization problems,'' *Appl. Intell.*, vol. 51, pp. 1531–1551, Sep. 2020.

[13] H. A. Alsattar, A. A. Zaidan, and B. B. Zaidan, ''Novel meta-heuristic bald eagle search optimisation algorithm,'' *Artif. Intell. Rev.*, vol. 53, no. 3, pp. 2237–2264, Mar. 2020.

[14] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, and H. Chen, ''Harris hawks optimization: Algorithm and applications,'' *Future Gener. Comput. Syst.*, vol. 97, pp. 849–872, Aug. 2019.

[15] M. Kumar, A. J. Kulkarni, and S. C. Satapathy, ''Socio evolution & learning optimization algorithm: A socio-inspired optimization methodology,'' *Future Gener. Comput. Syst.*, vol. 81, pp. 252–272, Apr. 2018.

[16] G. Dhiman and A. Kaur, ''STOA: A bio-inspired based optimization algorithm for industrial engineering problems,'' *Eng. Appl. Artif. Intell.*, vol. 82, pp. 148–174, Jun. 2019.

[17] G. Dhiman and V. Kumar, ''Spotted hyena optimizer: A novel bio-inspired based metaheuristic technique for engineering applications,'' *Adv. Eng. Softw.*, vol. 114, pp. 48–70, Dec. 2017.

[18] R. V. Rao, V. J. Savsani, and D. P. Vakharia, ''Teaching–learning-based optimization: An optimization method for continuous non-linear large scale problems,'' *Inf. Sci.*, vol. 183, no. 1, pp. 1–15, Jan. 2012.

[19] X. Jiang and S. Li, ''BAS: Beetle antennae search algorithm for optimization problems,'' *Int. J. Robot. Control*, vol. 1, no. 1, pp. 1–5, Apr. 2018.

[20] B. Chen, Z. Huang, R. Zhang, F. Jiang, W. Liu, H. Li, J. Wang, and J. Peng, ''Adaptive slip ratio estimation for active braking control of high-speed trains,'' *ISA Trans.*, vol. 112, pp. 302–314, Jun. 2021.

[21] Y. Sun, J. Zhang, G. Li, Y. Wang, J. Sun, and C. Jiang, ''Optimized neural network using beetle antennae search for predicting the unconfined compressive strength of jet grouting coalcretes,'' *Int. J. Numer. Anal. Methods Geomech.*, vol. 43, no. 4, pp. 801–813, Mar. 2019.

[22] Z. Liu, Q. Tan, Y. Zhou, and H. Xu, ''Syncretic application of IBAS-BP algorithm for monitoring equipment online in power system,'' *IEEE Access*, vol. 9, pp. 21769–21776, 2021.

[23] Z. Hu and P. Zhang, ''Damage identification of structures based on smooth orthogonal decomposition and improved beetle antennae search algorithm,'' *Adv. Civil Eng.*, vol. 2021, Feb. 2021, Art. no. 8857356.

[24] J. Zhang, G. Ma, Y. Huang, J. Sun, F. Aslani, and B. Nener, ''Modelling uniaxial compressive strength of lightweight self-compacting concrete using random forest regression,'' *Construct. Building Mater.*, vol. 210, pp. 713–719, Jun. 2019.

[25] Y. Xu, Y. Huang, and G. Ma, ''A beetle antennae search improved BP neural network model for predicting multi-factor-based gas explosion pressures,'' *J. Loss Prev. Process Ind.*, vol. 65, pp. 1–9, Oct. 2020.

[26] L. Ji, Z. Cao, Q. Hong, X. Chang, Y. Fu, J. Shi, Y. Mi, and Z. Li, ''An improved inverse-time over-current protection method for a microgrid with optimized acceleration and coordination,'' *Energies*, vol. 13, no. 21, p. 5726, Nov. 2020.

[27] X. Li, H. Jiang, M. Niu, and R. Wang, ''An enhanced selective ensemble deep learning method for rolling bearing fault diagnosis with beetle antennae search algorithm,'' *Mech. Syst. Signal Process.*, vol. 142, Aug. 2020, Art. no. 106752.

[28] J. Zhang, Y. Huang, G. Ma, J. Sun, and B. Nener, ''A Metaheuristic-optimized multi-output model for predicting multiple properties of pervious concrete,'' *Construct. Building Mater.*, vol. 249, Jul. 2020, Art. no. 118803.

[29] P. Wang, Y. Gao, M. Wu, F. Zhang, G. Li, and C. Qin, ''A denoising method for fiber optic gyroscope based on variational mode decomposition and beetle swarm antenna search algorithm,'' *Entropy*, vol. 22, no. 7, p. 765, Jul. 2020.

[30] A. T. Khan, X. Cao, S. Li, B. Hu, and V. N. Katsikis, ''Quantum beetle antennae search: A novel technique for the constrained portfolio optimization problem,'' *Sci. China Inf. Sci.*, vol. 64, no. 5, May 2021, Art. no. 152204.

[31] Q. Wu, X. Shen, Y. Jin, Z. Chen, S. Li, A. H. Khan, and D. Chen, ''Intelligent beetle antennae search for UAV sensing and avoidance of obstacles,'' *Sensors*, vol. 19, no. 8, P. 1758, 2019.

[32] A. Rajagopal, A. Ramachandran, K. Shankar, M. Khari, S. Jha, and G. P. Joshi, ''Optimal routing strategy based on extreme learning machine with beetle antennae search algorithm for low earth orbit satellite communication networks,'' *Int. J. Satell. Commun. Netw.*, vol. 39, no. 3, pp. 305–317, May 2021.

[33] M. Lin, Q. Li, F. Wang, and D. Chen, ''An improved beetle antennae search algorithm and its application on economic load distribution of power system,'' *IEEE Access*, vol. 8, pp. 99624–99632, 2020.

[34] Q. Wu, H. Lin, Y. Jin, Z. Chen, S. Li, and D. Chen, ''A new fallback beetle antennae search algorithm for path planning of mobile robots with collision-free capability,'' *Soft Comput.*, vol. 24, no. 3, pp. 2369–2380, Feb. 2020.

[35] X. Jiang, Z. Lin, T. He, X. Ma, S. Ma, and S. Li, ''Optimal path finding with beetle antennae search algorithm by using ant colony optimization initialization and different searching strategies,'' *IEEE Access*, vol. 8, pp. 15459–15471, 2020.

[36] S. Xie, V. Garofano, X. Chu, and R. R. Negenborn, ''Model predictive ship collision avoidance based on Q-learning beetle swarm antenna search and neural networks,'' *Ocean Eng.*, vol. 193, Dec. 2019, Art. no. 106609.

[37] S. Gao, Y. Zhang, Y. Zhang, and G. Zhang, ''Elman neural network soft-sensor model of PVC polymerization process optimized by chaos beetle antennae search algorithm,'' *IEEE Sensors J.*, vol. 21, no. 3, pp. 3544–3551, 2021.

[38] V. N. Katsikis, S. D. Mourtas, P. S. Stanimiroviá, S. Li, and X. Cao, ''Time-varying minimum-cost portfolio insurance under transaction costs problem via beetle antennae search algorithm (BAS),'' *Appl. Math. Comput.*, vol. 385, Nov. 2020, Art. no. 125453.

[39] Q. Wu, Z. Ma, G. Xu, S. Li, and D. Chen, ''A novel neural network classifier using beetle antennae search algorithm for pattern classification,'' *IEEE Access*, vol. 7, pp. 64686–64696, 2019.

[40] J. Yang and Z. Peng, ''Beetle-swarm evolution competitive algorithm for bridge sensor optimal placement in SHM,'' *IEEE Sensors J.*, vol. 20, no. 15, pp. 8244–8255, Aug. 2020.

[41] L. Zhou, K. Chen, H. Dong, S. Chi, and Z. Chen, ''An improved beetle swarm optimization algorithm for the intelligent navigation control of autonomous sailing robots,'' *IEEE Access*, vol. 9, pp. 5296–5311, 2021.

[42] X. Xu, K. Deng, and B. Shen, ''A beetle antennae search algorithm based on Lévy flights and adaptive strategy,'' *Syst. Sci. Control Eng.*, vol. 8, no. 1, pp. 35–47, Jan. 2020.

[43] L. Wang, Q. Wu, F. Lin, S. Li, and D. Chen, ''A new trajectory-planning beetle swarm optimization algorithm for trajectory planning of robot manipulators,'' *IEEE Access*, vol. 7, pp. 154331–154345, 2019.

[44] Y. Cheng, C. Li, S. Li, and Z. Li, ''Motion planning of redundant manipulator with variable joint velocity limit based on beetle antennae search algorithm,'' *IEEE Access*, vol. 8, pp. 138788–138799, 2020.

[45] A. H. Khan, S. Li, B. Xu, and X. Cao, ''A model-free approach for online optimization of nonlinear systems,'' *IEEE Trans. Circuits Syst. I, Reg. Papers*, early access, May 11, 2021, doi: 10.1109/TCSII.2021.3079125.

[46] Z. Yue, G. Li, X. Jiang, S. Li, J. Cheng, and P. Ren, ''A hardware descriptive approach to beetle antennae search,'' *IEEE Access*, vol. 8, pp. 89059–89070, 2020.

[47] T. Ghosh and K. Martinsen, ''A collaborative beetle antennae search algorithm using memory based adaptive learning,'' *Appl. Artif. Intell.*, vol. 35, no. 6, pp. 440–475, May 2021.

[48] A. T. Khan, X. Cao, Z. Li, and S. Li, ''Enhanced beetle antennae search with zeroing neural network for online solution of constrained optimization,'' *Neurocomputing*, vol. 447, pp. 294–306, Aug. 2021.

[49] X. Jiang, Z. Lin, S. Li, Y. Ji, Y. Luan, and S. Ma, ''Dynamical attitude iguration with wearable wireless body sensor networks through beetle antennae search strategy,'' *Measurement*, vol. 167, Jan. 2021, Art. no. 108128.

[50] Z. Wang, L. Yao, Y. Cai, and J. Zhang, ''Mahalanobis semi-supervised mapping and beetle antennae search based support vector machine for wind turbine rolling bearings fault diagnosis,'' *Renew. Energy*, vol. 155, pp. 1312–1327, Aug. 2020.

[51] H. Zhang, Z. Li, X. Jiang, X. Ma, J. Chen, S. Li, Y. Luan, Z. Lv, and S. Ma, ''Beetle colony optimization algorithm and its application,'' *IEEE Access*, vol. 8, pp. 128416–128425, 2020.

[52] A. T. Khan, S. Li, and X. Zhou, ''Trajectory optimization of 5-link biped robot using beetle antennae search,'' *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 68, no. 10, pp. 3276–3280, Oct. 2021.

[53] Z. Liao, X. Song, B. Jia, and P. Chen, ''Automatic bearing fault feature extraction method via PFDIC and DBAS,'' *Math. Problems Eng.*, vol. 2021, May 2021, Art. no. 6655081.

[54] J. Huang, G. Shiva Kumar, J. Ren, J. Zhang, and Y. Sun, ''Accurately predicting dynamic modulus of asphalt mixtures in low-temperature regions using hybrid artificial intelligence model,'' *Construct. Building Mater.*, vol. 297, Aug. 2021, Art. no. 123655.

[55] Y. Zhang, S. Li, and B. Xu, ''Convergence analysis of beetle antennae search algorithm and its applications,'' 2019, *arXiv:1904.02397*. [Online]. Available: http://arxiv.org/abs/1904.02397

[56] L. Liao and H. Yang, ''Review of beetle antennae search,'' *Comput. Eng. Appl.*, vol. 57, no. 12, pp. 54–64, 2020.

[57] J. Zou, Q. Deng, J. Zheng, and S. Yang, ''A close neighbor mobility method using particle swarm optimizer for solving multimodal optimization problems,'' *Inf. Sci.*, vol. 519, pp. 332–347, May 2020.

[58] X.-S. Yang and A. H. Gandomi, ''Bat algorithm: A novel approach for global engineering optimization,'' *Eng. Comput.*, vol. 29, no. 5, pp. 464–483, 2012.

[59] Y. G. Xue and H. W. Deng, ''The cuckoo search algorithm based on dynamic grouping to adjust flight scale,'' *Appl. Mech. Mater.*, vols. 543–547, pp. 1822–1826, Mar. 2014.

[60] I. H. Osman, ''Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem,'' *Ann. Oper. Res.*, vol. 41, no. 4, pp. 421–451, 1993.

[61] G.-G. Wang, L. H. Guo, A. H. Gandomi, G.-S. Hao, and H. Q. Wang, ''Chaotic krill herd algorithm,'' *Inf. Sci.*, vol. 274, pp. 17–34, Aug. 2014.

[62] Z. Cui, F. Li, and W. Zhang, ''Bat algorithm with principal component analysis,'' *Int. J. Mach. Learn. Cybern.*, vol. 10, no. 3, pp. 603–622, Mar. 2019.

[63] J. Rueda and I. Erlich, ''Hybrid population based MVMO for solving CEC 2018 test bed of single-objective problems,'' in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jun. 2018, pp. 1–8.

[64] S. Arora and S. Singh, ''Butterfly optimization algorithm: A novel approach for global optimization,'' *Soft Comput.*, vol. 23, no. 3, pp. 715–734, 2018.

[65] S. Mirjalili, S. M. Mirjalili, and A. Lewis, ''Grey wolf optimizer,'' *Adv. Eng. Softw.*, vol. 69, pp. 46–61, Mar. 2014.

[66] Z. Xing and H. Jia, ''Multilevel color image segmentation based on GLCM and improved Salp Swarm algorithm,'' *IEEE Access*, vol. 7, pp. 37672–37690, 2019.

[67] S. Mirjalili, ''SCA: A sine cosine algorithm for solving optimization problems,'' *Knowl.-Based Syst.*, vol. 96, pp. 120–133, Mar. 2016.

[68] S. Mirjalili, A. H. Gandomi, S. Z. Mirjalili, S. Saremi, H. Faris, and S. M. Mirjalili, ''Salp swarm algorithm: A bio-inspired optimizer for engineering design problems,'' *Adv. Eng. Softw.*, vol. 114, pp. 163–191, Dec. 2017.

[69] J. Wu, R. Nan, and L. Chen, ''Improved salp swarm algorithm based on weight factor and adaptive mutation,'' *J. Exp. Theor. Artif. Intell.*, vol. 31, no. 3, pp. 493–515, May 2019.

[70] S. Mirjalili and A. Lewis, ''The whale optimization algorithm,'' *Adv. Eng. Softw.*, vol. 95, pp. 51–67, May 2016.

[71] G. Zhang and Y. Shi, ''Hybrid sampling evolution strategy for solving single objective bound constrained problems,'' in *Proc. IEEE Congr. Evol. Comput. (CEC)*, 2018, pp. 1–7.

XUAN SHAO received the bachelor's degree from Beijing University of Aeronautics and Astronautics, in 2008, the master's degree from The University of Manchester, in 2010, and the Ph.D. degree from Harbin University of Science and Technology, in 2018. She has been a Lecturer with Harbin University of Science and Technology. Her main research interests include intelligent algorithm and intelligent control.

YUQI FAN received the bachelor's degree in mechanical engineering and automation and the master's degree in mechanical and electronic engineering from Harbin University of Science and Technology, in 2014 and 2017, respectively, where she is currently pursuing the Ph.D. degree. Her current research interests include intelligent algorithm and intelligent control.