

Received April 18, 2022, accepted May 18, 2022, date of publication May 23, 2022, date of current version May 31, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3177218

A Dual Biogeography-Based Optimization Algorithm for Solving High-Dimensional Global Optimization Problems and Engineering Design Problems

ZIYU ZHANG^{1,2}, YUELIN GAO^{1,2}, AND WENLU ZUO¹

¹School of Mathematics and Information Science, North Minzu University, Yinchuan 750021, China

²Ningxia Province Key Laboratory of Intelligent Information and Data Processing, Yinchuan 750021, China

Corresponding author: Yuelin Gao (gaoyuelin@263.net)

This work was supported in part by the Key Project of Ningxia Natural Science Foundation Several Swarm Intelligence Algorithms and Their Application under Grant 2022AAC02043, in part by the 2022 Graduate Innovation Project of North Minzu University under Grant YCX22095, in part by the National Natural Science Foundation of China under Grant 11961001, in part by the Construction Project of First-Class Subjects in Ningxia Higher Education under Grant NXYLXK2017B09, and in part by the Major Proprietary Funded Project of North Minzu University under Grant ZDZX201901.

ABSTRACT Biogeography-based optimization (BBO) cannot effectively solve high-dimensional global optimization problems due to its single migration mechanism and random mutation operator. To get better performance, a dual BBO based on sine cosine algorithm (SCA) and dynamic hybrid mutation is proposed in this work, which named SCBBO. Firstly, the Latin hypercube sampling method is innovatively used to improve the initial population ergodicity. Secondly, a nonlinear transformation parameter and a inertia weight adjustment factor are designed into the position update formula of SCA to make SCBBO suitable for high dimensional environments. Then, a dynamic hybrid mutation operator is designed by combining Laplacian and Gaussian mutation, which helps the algorithm to escape from local optima and balance the exploration and exploitation. Finally, the dual learning strategy is integrated, so the convergence accuracy is further improved by generating dual individuals. Meanwhile, A sequence convergence model is established to prove the algorithm can converge to the global optimal solution with probability 1. Compared with other state-of-the-art evolutionary algorithms, SCBBO effectively improves the optimization accuracy and convergence speed for high-dimensional optimization problems. To further show the superiority of SCBBO, its performance is compared on 1000, 2000, 5000 and 10000 dimensions, respectively. The comparisons show that SCBBO's optimization results on these dimensions are basically the same. Applying SCBBO to engineering design problems, and the simulation results demonstrate that the proposed method is also effective on constrained optimization problems.

INDEX TERMS Biogeography-based optimization, high-dimensional global optimization, sine cosine algorithm, dual learning, engineering design problems.

I. INTRODUCTION

Optimization is the most common problem in engineering, science, economy and society, such as system control, mechanical design, network design, large-scale integrated circuit design, biopharmaceutical and economic model. The application of optimization technology in the above-mentioned domains has produced enormous economic and

social benefits. Practice shows that, under the same conditions, the system efficiency, resources, economic benefits and other aspects have been significantly improved through the treatment of optimization technology, and the larger the scale of the problem, the more obvious the corresponding effect. With the development of engineering technology and science, optimization problems tend to be large-scale, multi-peak, nonlinear and strongly constrained. The objective function is discontinuous and non-differentiable, and some of them even have no clear function form. Traditional optimization

The associate editor coordinating the review of this manuscript and approving it for publication was Hiram Ponce.

methods, such as Newton method, quasi-Newton method, conjugate gradient method, variable scale method and tunnel method, are no longer applicable due to the following problems: *a*). The traditional method requires continuous and differentiable objective function. *b*). Before the algorithm is implemented, the first-order or even higher-order derivatives and inverse matrices of the function are required. The more complex the objective function is, the more calculation is required. *c*). The results of the problem are closely related to the selection of initial values. *d*). The algorithm lacks generality, and users need to select the most appropriate method for specific problems. In order to solve the above problems, many scholars have been inspired by nature and designed various algorithms based on population by imitating biological mechanisms or natural phenomena, which are called swarm intelligence evolutionary algorithms (EAs). EAs are not constrained by the restriction conditions (such as differentiability, continuity etc.) and do not need derivative and other auxiliary information. They have the characteristics of high efficiency, simple operation, strong universality and meets the objective requirements of current optimization problems. Therefore, EAs have become the mainstream optimization method.

In the past 40 years, swarm intelligence EAs have developed rapidly and have been widely used in communication [1], finance [2], power grid [3], military [4], control system [5], [6] and other fields. Among them, the biogeography-based optimization (BBO) proposed by Dr. Simon of the United States in 2008 is a new swarm heuristic algorithm [7]. Simon uses the mechanism of biological migration and information interaction between habitats to establish mathematical modeling to realize the optimization process of the problem. BBO has the advantages of few parameters, easy implementation, and good stability when searching for the global optimal solution. So it has been widely favored by scholars from all over the world since it was proposed. Up to now, there are still many scholars doing in-depth research on the algorithm. They have improved or applied it to practical problems, and achieved some results.

In the field of algorithm improvement, people mainly make improvements in three directions. The first direction is to adjust the migration rate model of BBO. Ma [8] designed six migration rate models and found that the cosine migration model has the best optimization performance, and concluded that the performance of the nonlinear migration model is far better than that of the linear migration model. Wei *et al.* [9] inspired by the cosine migration model and designed a more complex hyperbolic sine cosine migration model, which further improves the optimization performance of BBO. The second direction is to improve the operator of BBO. BBO's framework is mainly composed of three operators: selection, migration and mutation. For the selection operator, Feng *et al.* [10] designed a selection operator with a random ring topology in 2017, which reduces the possibility of better solutions being destroyed by inferior solutions. Later,

Zhang *et al.* [11] deleted the original BBO roulette selection operator in 2019, and adopted a example learning method to select better habitats for migration, thereby improving the convergence accuracy of the algorithm. An *et al.* [12] put forward a probabilistic selection operator based on non-dominated sorting in 2021, so that BBO can effectively solve the multi-objective flexible job-shop scheduling problem. For the migration operator, Literature [13] designed an enhanced biogeography-based optimization (BBO) referred to as POLBBO. In POLBBO, an efficient operator named polyphyletic migration operator is proposed. This operator can not only generate new features from more promising areas in the search space, but also effectively increase the population diversity. Then, Bansal *et al.* [14] proposed a new operator, namely the disruption operator to improve the capability of exploration and exploitation in BBO. This new algorithm is called DisruptBBO (DBBO). literature [15] designed a disturbed migration operator and obtained the PBBO. PBBO increases the local development capabilities of BBO. Literature [16] designed a novel BBO by integrating opposition-based learning mechanism (OBBO). In OBBO, the opposite individuals are merged into BBO population to improve the diversity, and the optimization performance of this algorithm is obviously better than that of standard BBO. Recently, Reihanian *et al.* [17] introduced a new two-stage migration operator in the framework of BBO to enable the algorithm to search the problem space effectively. For the mutation operator, the harmony search (HS) [19] process was added to the mutation operator of BBO in literature [18], and HSBBO was obtained. HSBBO not only effectively increases population diversity, but also improves the convergence accuracy. Zheng *et al.* [20] directly deleted the random mutation operator of BBO and adopted the differential mutation mechanism to conduct effective search, thus improving the algorithm's ability to develop new solutions. The last direction of improvement is to integrate BBO with other EAs. Literature [21] presented a biogeography-based krill herd (BBKH) algorithm to solve complex optimization problems. Literature [22] proposed the BBOTS based on tabu search algorithm (TS) [23]. It stores the performed migrations in a taboo table and prohibits reverse migration of populations to previous habitats. Yogesh *et al.* [24] integrated particle swarm optimization (PSO) [25] into BBO, and applied it to speech signal and emotional stress recognition. Zhang *et al.* [26] presented a novel hybrid algorithm based on BBO and grey wolf optimizer (GWO) [27], named HBBOG. In recent two years, with the continuous emergence of new EAs, many scholars have integrated some new algorithms into BBO. Farswan *et al.* [28] fused fireworks algorithm (FWA) [29] with BBO to obtain the FBBO, which has two search mechanisms. Then, Hamid [30] merged the firefly algorithm (FA) [31] with BBO in 2021, and obtained the hybrid algorithm FABBO. FABBO is essentially a two-stage method. In the first stage, FA is used for preliminary optimization, and some better solutions are found by searching the problem space through a limited iteration.

In the second stage, BBO is used to conduct more refined search for these better solutions to obtain the optimal solution with higher accuracy.

In the field of algorithm application, BBO has been applied in many aspects at present. For example, literature [32] solved the function optimization problem of discrete variables by improving BBO. Literature [33] proposed a novel BBO based on population competition strategy to solve the substation location problem. Then, Pal and Saraswat [34] introduced an innovative method for categorization of histopathological images using an enhanced bag-of-feature framework. To obtain the optimal visual words in bag-of-features, they proposed a new spiral BBO variant which introduces a spiral search and random search in the mutation operator to generate the suitability index variables. By 2021, BBO has been applied to fields such as industrial production and financial optimization. For instance, Rostami *et al.* [35] designed an optimal feature selection method for SAR image classification based on BBO, artificial bee colony (ABC) and support vector machine (SVM) in order to solve the feature selection problem. Harrabi *et al.* [36] designed a hybrid BBO algorithm to solve job-shop scheduling problems with general time delays. Literature [37] also proposed a hybrid meta-heuristic method based on BBO and PSO to estimate the currency demand in Iran. And Recently, Taghizadeh *et al.* [38] proposed a metaheuristic-based data replica placement mechanism using BBO for data-intensive IoT applications on the fog ecosystem.

Although BBO is easy to implement and has few parameters, it is easy to fall into local optimal solution and difficult to escape [39]. Especially in the late evolution, the convergence rate of BBO is very slow. The improvement of scholars from various countries has reduced the possibility of BBO falling into local optimal solution, and the algorithm has been improved to a certain extent [40]. However, the convergence speed and optimization accuracy of BBO still need to be improved, especially in the late evolution, the search speed has not been effectively improved. However, according to our extensive investigation, the present BBO and its variants are not effective in solving high-dimensional optimization problems. We review hundreds of studies on BBO in the last decade and find that none of them solved problems with more than 100 dimensions. The vast majority of variants have only been tested on 30, 50 or 100 dimensions. With the progress of society, practical problems put forward higher and higher requirements for algorithms. In order to achieve a breakthrough in this field, aiming at the shortcomings of BBO, this paper proposes a dual BBO with sine cosine algorithm (SCA) [65] and dynamic hybrid mutation mechanism. As it integrates another evolutionary algorithm, it mainly belongs to the third category in the direction of improvement, so we named it SCBBO. SCBBO improves the original algorithm migration operator and mutation operator respectively, and integrates SCA and dual learning strategy. The above improved methods make BBO adapt to high dimensional optimization environment, and

greatly improve the convergence performance of BBO. The main contributions of this paper are as follows:

(1). In this paper, SCA algorithm and BBO algorithm are innovatively combined to obtain a hybrid migration algorithm. At the same time, a dynamic hybrid mutation operator is designed to effectively balance the exploration and exploitation of the algorithm. In addition, this paper designs a dual learning strategy and combines it into BBO for the first time. A sequence convergence model is established to prove the convergence of SCBBO. This is a new proof.

(2). The ability of SCBBO to solve global optimization problems can reach 10000 dimensions. To further show the superiority of SCBBO on high dimensions, we test SCBBO's optimization ability on 1000, 2000, 5000, 10000 dimensions. The results prove that the algorithm in this paper has good stability and excellent optimization ability on high-dimensional environment.

(3). SCBBO is applied to solve the engineering design optimization. It solves pressure vessel design, tension/compression spring design and welded beam design respectively. By comparing the results of other literatures and algorithms, we conclude that that SCBBO has better applicability and optimization capabilities in engineering design problems.

The remaining sections of this paper are as follows: section II introduces the standard BBO and its calculation process in detail; In section III, the algorithm SCBBO is proposed. In section IV, the global convergence of SCBBO is proved. In section V, the computational complexity of SCBBO is analyzed. Then, section VI includes numerical experiments and results analysis. section VII is the application of SCBBO, using SCBBO to solve engineering design optimization problems. Finally, section VIII summarizes the whole work and points out the direction of future work. The graphical abstract of this paper is shown in FIGURE 1.

II. STANDARD BBO

Simon proposed the BBO in 2008. The basic idea is that biological populations live in different habitats and are affected by rainfall, vegetation diversity, geological diversity, climate and so on. The suitability of each habitat is different, and biological populations are distributed and migrated accordingly. In an optimization problem, a habitat corresponds to a candidate solution, and the habitat suitability index (HSI) corresponds to the objective function value of the candidate solution. The aforementioned factors affecting HSI are called suitability index variables (SIVs), which correspond to independent variables of candidate solutions. If the candidate solutions are considered as individuals in the population, the good individuals are like the habitats with high HSI, and the bad individuals are like the habitats with low HSI. Good individuals are more likely to share their independent variables with poor individuals, and poor individuals are more likely to accept the characteristics of good individuals. The addition of new features may improve the quality of individuals, and obtain a better function

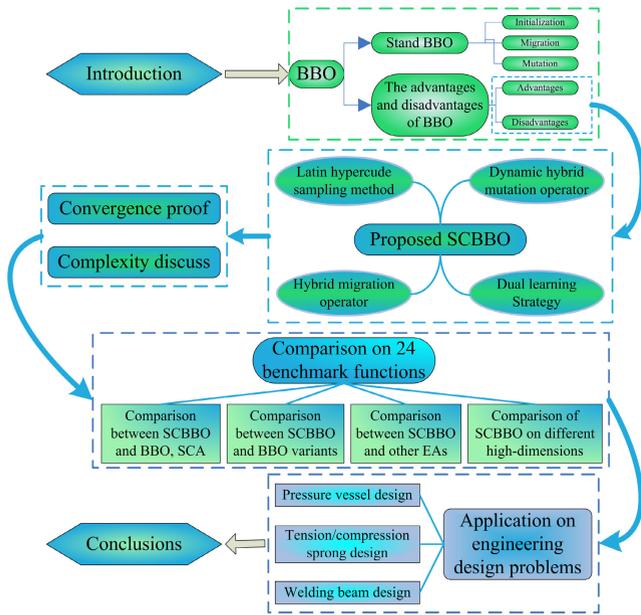


FIGURE 1. Graphical abstract of this paper.

target value, which is the mathematical idea of BBO [39], [40]. BBO is an EA, which is mainly accomplished by the following three steps.

A. INITIALIZATION

BBO uses Eq. (1) to randomly generate NP habitats as the initial population, and each habitat contains D variables.

$$x_{ij} = x_{j\min} + \text{rand}(0, 1) \cdot (x_{j\max} - x_{j\min}) \quad (1)$$

where, $i = 1, 2, \dots, NP$; $j = 1, 2, \dots, D$. x_{ij} is the j -th variable of habitat x_i , so $x_i = (x_{i1}, x_{i2}, \dots, x_{iD})$. $x_{j\max}$ and $x_{j\min}$ are the upper and lower limits of the j -th variable respectively. After population initialization, HSI of each habitat can be calculated based on fitness function of the actual problem.

B. MIGRATION

The population is sorted in descending order by the HSI of each x_i , so the original habitat x_i will be assigned a new i . The species number S_i of habitat x_i sorted can be calculated using Eq. (2):

$$S_i = S_{\max} - i, \quad i = 1, 2, \dots, NP \quad (2)$$

where, S_{\max} is the maximum number of species and is given as an initial value. Immigration rate and emigration rate are calculated based on the species number of habitats. Habitats with high HSI have a high probability of sharing features with habitats with low HSI to improve the quality of those low HSI habitats. At the same time, habitats with high HSI will resist change, so their immigration rate is low [7]. Therefore, habitats with high HSI have higher emigration rates and lower immigration rates than habitats with low HSI, while habitats with low HSI have the opposite results. In general, the migration process follows the migration rate model.

The original BBO used the linear migration model to calculate the immigration and emigration rate, but literature [8] proved that complex migration model has better optimization performance than linear model. So, this paper adopts the cosine migration model which is more consistent with the natural law. Compared with linear migration model, it can better reflect the nature of ecosystem migration and increase species diversity. Therefore, the immigration rate λ_i and emigration rate μ_i of habitat x_i are calculated by Eq. (3):

$$\lambda_i = \frac{I}{2} \left(\cos \frac{\pi \cdot S_i}{S_{\max}} + 1 \right), \mu_i = \frac{E}{2} \left(-\cos \frac{\pi \cdot S_i}{S_{\max}} + 1 \right). \quad (3)$$

where, I is the maximum immigration rate and E is the maximum emigration rate, both of which are given as initial values.

For each habitat x_i , the characteristic variables to be immigrated should be determined according to λ_i during the migration process. The specific operation is to generate a random number between (0,1) for each variable of habitat x_i . If it is smaller than λ_i , this variable needs to be replaced. Then, in the remaining $NP-1$ habitats, the habitat x_k to be emigrated is determined by roulette according to μ_k . Finally, the variable of x_k is used to replace the corresponding variable of x_i . Algorithm 1 shows the BBO migration process.

Algorithm 1 The Migration Operator of BBO

```

for  $i = 1$  to  $NP$ 
  for  $j = 1$  to  $D$ 
    if  $\text{rand}(0,1) < \lambda_i$ 
      select the  $x_k$  according to the  $\{\mu_k\}_{k=1}^{NP}$ 
       $x_{ij} = x_{kj}$ 
    end if
  end for
end for
    
```

C. MUTATION

Catastrophic events (e.g. famine, natural disaster, etc.) that suddenly changes the HSI of a habitat, or an outbreak that causes a species to move to another habitat, or a genetic mutation that directly creates a new species, are all referred to as mutation. Firstly, the species probability P_i of each habitat is calculated from the immigration rate λ_i and emigration rate μ_i through Eq. (4).

$$P_i = \begin{cases} -(\lambda_i + \mu_i) P_i + \mu_{i+1} P_{i+1}, & S_i = 0 \\ -(\lambda_i + \mu_i) P_i \\ +\lambda_{i-1} P_{i-1} + \mu_{i+1} P_{i+1}, & 1 \leq S_i \leq S_{\max} - 1 \\ -(\lambda_i + \mu_i) P_i + \lambda_{i-1} P_{i-1}, & S_i = S_{\max} \end{cases} \quad (4)$$

The mutation rate of a habitat is inversely proportional to its species probability [39]. Therefore, the relationship between the mutation rate m_i and the species probability P_i

of each habitat is as follows:

$$m_i = \left(1 - \frac{P_i}{P_{\max}}\right) \cdot m_{\max}, P_{\max} = \max\left(\{P_i\}_{i=1}^{NP}\right) \quad (5)$$

where, m_{\max} is the maximum mutation rate, which is given as an initial value. For each habitat x_i , a number between (0,1) is randomly generated, and if it is smaller than the mutation rate m_i , x_i needs to be mutated. Then for each independent variable of x_i , a random number in the range of values is generated to replace the original variable value. Algorithm 2 gives the mutation process of BBO.

Algorithm 2 The Mutation Operator of BBO

```

for  $i = 1$  to  $NP$ 
  if  $\text{rand}(0,1) < m_i$ 
    for  $j = 1$  to  $D$ 
      updata  $x_{ij}$  by Eq. (1)
    end for
  end if
end for

```

D. THE CALCULATION PROCESS OF BBO

Algorithm 3 presents the detailed calculation process of BBO.

E. THE ADVANTAGES AND DISADVANTAGES OF BBO

Since the main work of this paper is to use BBO to solve high dimensional global optimization problems, we mainly discuss the advantages of using BBO to solve high dimensional numerical problems. Through the analysis of BBO to explain our motivation to improve BBO. Then, we analyze the reasons why the present BBO is not suitable for the high-dimensional optimization environment, so as to point out the direction for the next improvements.

The advantages of BBO:

(1). Unlike other evolutionary algorithms (e.g. genetic algorithm, differential evolution algorithm, immune algorithm), BBO does not need to breed or cross to produce the next generation of population, so it can greatly reduce the complexity of the algorithm. Even if solving high-dimensional numerical problems, it does not consume too much memory.

(2). BBO does not require complex parameter tuning like particle swarm optimization, ant colony optimization or artificial bee colony algorithm. BBO's parameters are basically fixed and do not need to be reset according to the nature of the problem. Therefore, the parameters do not affect the convergence performance of the algorithm when solving high-dimensional optimization problems.

(3). BBO has good utilization ability of population information. It uses information from current population to migrate species and evolve. Therefore, on the high-dimensional environment, the population can still complete feature sharing in all dimensions, so as to evolve towards the optimal solution.

The disadvantages of BBO:

(1). According to Eq. (1), BBO uses random initialization to generate initial population. This method makes it difficult to disperse the population in high dimensional space. The population has no ergodicity, so the algorithm converges slowly on solving high dimensional problems.

(2). BBO uses roulette to select habitats for emigration. It cannot avoid the immigration of inferior individuals to superior individuals. If habitat x_i will be immigrated, it is likely to be immigrated by habitat $x_j (j > i)$. It means that habitats with lower HSI will immigrate to habitats with higher HSI, and the features of poorer individuals will replace the features of better individuals, thus reducing the fitness of superior individuals. Therefore, the random selection of BBO will reduce population diversity and thus not suitable for high-dimensional environment.

(3). BBO's search capability is weak. BBO searches the problem space using information sharing between species. This mechanism works well in low-dimensional environment. However, when solving high-dimensional problems, a large number of new individuals are needed to search the space, and only the new solution generated by the migration operator is far from enough.

(4). BBO uses random mutation to escape from the local optima. However, for the individuals with high fitness, random mutation can easily destroy them, leading to worse individuals and lower population diversity. This mutation method is blind and cannot guarantee the mutation to the direction of the optimal solution. But, when solving high-dimensional optimization problems, it is difficult to find the global optimal solution if the search is blind.

(5). BBO can not balance the exploration and exploitation effectively. It only relies on the substitution of several variables to search the problem space. Therefore, the algorithm cannot effectively switch between local search and global search in high-dimensional space.

III. PROPOSED ALGORITHM (SCBBO)

Although many variants of BBO have been put forward by scholars from various countries, these variants still have many defects on solving some complex problems. Especially for the high-dimensional global optimization problems, there is no variant can solve them effectively. In this section, some existing BBO variants are deeply studied, and the defects of different variants are analyzed. According to the defects of BBO and its variants, a dual BBO based on sine cosine algorithm and dynamic hybrid mutation mechanism is proposed and named SCBBO. We will introduce the design principle and calculation process of SCBBO in detail.

A. LATIN HYPERCUBE SAMPLING METHOD

The convergence speed and accuracy of the algorithm will be affected by the quality of the initial population. From subsection II-E, The initial population of standard BBO is randomly generated, so the diversity and the rationality of the distribution in the search space cannot be

Algorithm 3 The Calculation Procedure of BBO

```

initialize BBO parameters, which involve  $S_{max}$ ,  $I$ ,  $E$ ,  $NP$ , and  $m_{max}$ 
initialize the population of BBO by Eq. (1)
while (the termination condition is not met)
    for each habitat, calculate the HSI and sort from best to worst
    for each habitat, calculate the  $S_i$  by Eq. (2), the  $\lambda_i$  and  $\mu_i$  by Eq. (3)
    for each habitat, calculate the  $P_i$  by Eq. (4), the  $m_i$  by Eq. (5)
    perform the migration operator by Algorithm 1
    perform the mutation operator by Algorithm 2
end while
output the optimal solution

```

guaranteed. Most BBO variants use random initialization to generate initial populations, which is the main reason why they are not suitable for high-dimensional environment. At present, the improvement of population initialization strategy mainly uses chaotic mapping [41], [42]. However, chaotic mapping can only reduce the number of individuals distributed in the edge region of the search space, but can not effectively improve the ergodicity, and will increase the amount of computation. In this paper, the Latin hypercube sampling method is introduced to generate more uniform distribution of initial points without additional calculation.

Latin hypercube sampling is a multi-dimensional stratification technique that can efficiently sample in the distribution interval of variables [43]–[45]. The essence of this method is to divide the interval into N equally spaced non-overlapping sub-intervals, and conduct independent equal probability sampling for each sub-interval, so as to ensure that the sampling points are evenly distributed in the whole interval. Taking the interval $[0,1]$ as an example, random sampling and Latin hypercube sampling are carried out respectively. In the case of small sample numbers, random distribution does not disperse the population well to the whole interval. The comparison of the distribution of Latin hypercube sampling and random sampling is shown in FIGURE 2, in which 10 points are extracted from the interval $[0,1]$ throughing the two methods. It can be observed that Latin hypercube sampling can also spread over the entire space for a small number of samples. Therefore, this paper uses Latin hypercube sampling method to generate the initial population to improve the ergodicity.

B. HYBRID MIGRATION OPERATOR BASED ON SCA

Migration strategy has great influence on the search performance of BBO. Although the discrete migration mechanism of BBO can effectively utilize the current population information, direct substitution of solution variables will lead the blindness of migration. The variable of the better solution is likely to be replaced by the inferior solution, thus reducing the quality of the population, leading to the poor ability of the algorithm to mine new solutions. Moreover, the BBO variants proposed by other scholars still have some drawbacks.

For example, although the PRBBO [10] reduces the possibility that the better solution destroyed by the inferior solution, the selection operator of the random ring topology makes migration only take place between adjacent habitats, which largely reduces the ability of utilize population information. EMBBO [11] adopts the example learning method to select a better habitat than the current habitat for migration, thus improving the convergence accuracy of the algorithm. However, the whole population moves towards the local optimal solution, which is prone to fall into the local optima. TDBBO [46] designs a two-stage differential migration operator, which effectively balances the search and development capabilities of the algorithm, but has defects of high computational complexity and slow convergence. Therefore, in order to enhance the search ability, a hybrid migration operator based on sine cosine algorithm is proposed in this paper.

1) IMMIGRATION REFUSAL MECHANISM

From subsection II-E, the standard BBO is easy to migrate the features of the inferior solution to the superior one, so that the superior habitat is destroyed. In order to avoid the destruction of the inferior solution to the superior one, we design an immigration refusal mechanism. The specific operation is to set a threshold τ for the emigration rate μ_k of habitat x_k . When the emigration rate μ_k of habitat x_k is less than the threshold τ , habitat x_i will reject the variables from habitat x_k . The emigration rate of each habitat is proportional to its HSI. Therefore, the higher the emigration rate, the higher the HSI of the habitat, that is, the better the objective function value. Setting a threshold τ can ensure that all habitats used for emigration have high HSI, thus avoiding the destruction of superior habitats. Taking a population with only 6 individuals as an example, FIGURE 3 shows a sample graph of two migration operators.

As shown in FIGURE 3, threshold τ is similar to a dividing line. Habitats x_1, x_2, x_3 with an emigration rate greater than τ can share variables with other habitats, while habitats x_4, x_5, x_6 with an emigration rate less than τ can only accept variables from other habitats and cannot share information.

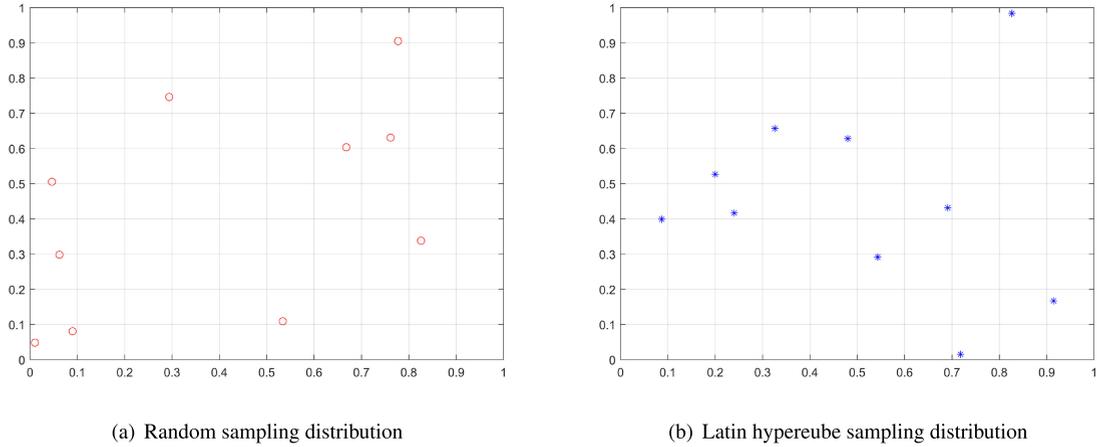


FIGURE 2. Comparison chart of sampling distribution.

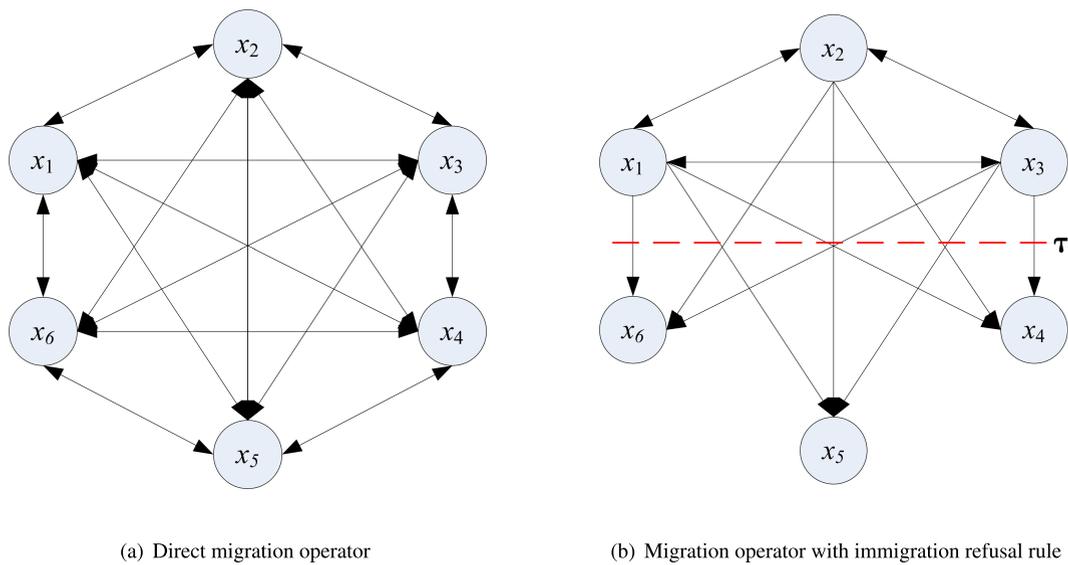


FIGURE 3. Direct migration operator and migration operator with immigration refusal mechanism.

2) CONVEX MIGRATION OPERATOR

The immigration refusal mechanism effectively avoids the damage of inferior solutions to the better ones, but does not greatly improve the convergence speed and accuracy of BBO. Therefore, when the emigration rate μ_k of habitat x_k selected by roulette is less than the threshold τ , a convex migration operator is adopted for the migration of habitat x_i . The features of x_i no longer just copy the features of x_k , but are replaced by a “convex combination” of x_k and the optimal solution of the current population x_{best} :

$$x_{ij} = (1 - \theta) \cdot x_{kj} + \theta \cdot x_{bestj}, \quad \theta \in \text{rand}(0, 1). \quad (6)$$

There are three reasons for adopting convex migration. Firstly, good individuals are less likely to degenerate as a result of migration, because some of their original characteristics are retained during migration. Secondly, poor

individuals will accept at least part of the characteristics from good individuals in the migration. Finally, such migration ensures that the population evolves towards the direction of the optimal value of each generation, no longer blindly searches, and can converge quickly. The parameter θ can be either deterministic or dynamically changing. Through a large number of experiments, this paper proposes the strategy of changing parameter θ dynamically and randomly. Because the current optimal solution x_{best} is likely to be the local optimal solution, dynamic random adjustment of θ can improve the probability of the algorithm escaping from the local optima.

3) SINE-COSINE MIGRATION OPERATOR

From subsection II-E, BBO only migrates the features of habitats whose random number is less than the immigration

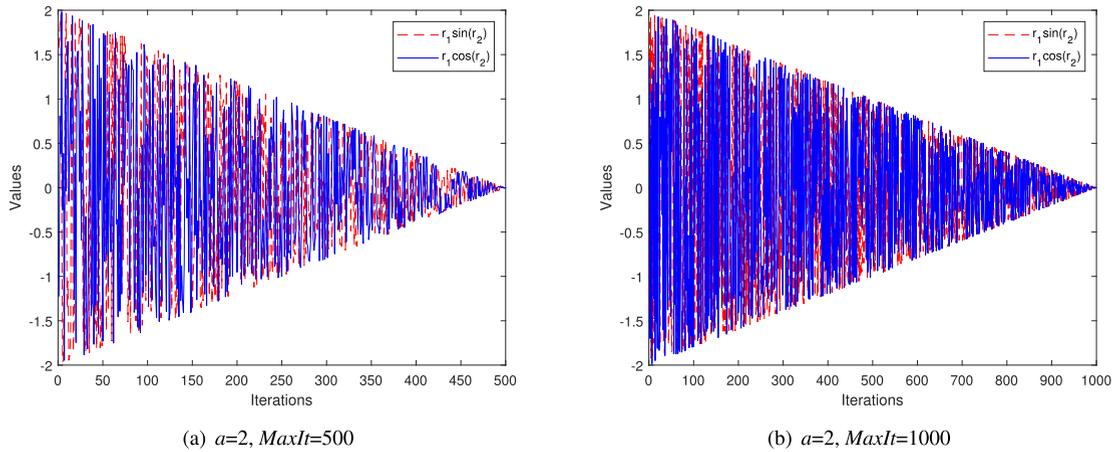


FIGURE 4. Images of $r_1 \sin(r_2)$ and $r_1 \cos(r_2)$.

rate, so its search capability is weak. The features of most habitats remain unchanged, so the population diversity does not increase significantly, which is also the main reason why the algorithm is not suitable for high-dimensional optimization environment. Therefore, this paper proposes a method of migration in the form of “sine and cosine function waves”, which together with immigration refusal mechanism and convex migration operator constitutes the hybrid migration operator.

The sine cosine algorithm (SCA) [47] has a simple structure and is easy to implement. Its most significant feature is that it achieves optimization based on the change of the value of sine function and cosine function. Each individual in SCA is updated through Eq. (7):

$$x_{ij}^{t+1} = \begin{cases} x_{ij}^t + r_1 \sin(r_2) \cdot |r_3 x_{bestj}^t - x_{ij}^t|, & \text{if } r_4 < 0.5 \\ x_{ij}^t + r_1 \cos(r_2) \cdot |r_3 x_{bestj}^t - x_{ij}^t|, & \text{if } r_4 \geq 0.5. \end{cases} \quad (7)$$

where, t is the current iteration number, $r_1 = a - at/MaxIt$, $r_2 \in [0, 2\pi]$, $r_3 \in [0, 2]$, $r_4 \in [0, 1]$, and $MaxIt$ is the maximum iteration number. In order to take full advantage of SCA’s search capabilities, we analyzed and discussed SCA in depth. In SCA, there are mainly four parameters (r_1, r_2, r_3, r_4). Among them, the most critical is the adaptive parameter r_1 , which controls the transformation of the algorithm from global search to local development. When the value of r_1 is large, the algorithm tends to search globally. When r_1 is small, the algorithm tends to local develop. Therefore, SCA uses the periodicity of sine and cosine for global search and local development. FIGURE 4 shows the graphs of $r_1 \sin(r_2)$ and $r_1 \cos(r_2)$ when $a = 2$, $MaxIt = 500$ and $a = 2$, $MaxIt = 1000$.

It can be seen from FIGURE 4 that when $r_1 > 1$, the values of $r_1 \sin(r_2)$ and $r_1 \cos(r_2)$ may be greater than 1 or smaller than -1. When $r_1 \leq 1$, the values of $r_1 \sin(r_2)$ and $r_1 \cos(r_2)$ must be between -1 and 1. According to SCA design principle, the algorithm performs global search first and then local search. The volatility of $r_1 \sin(r_2)$ and $r_1 \cos(r_2)$ and

their corresponding relationship with the algorithm search strategy as shown in FIGURE 5. When $|r_1 \sin(r_2)| > 1$ or $|r_1 \cos(r_2)| > 1$, the algorithm performs global search. When $|r_1 \sin(r_2)| \leq 1$ or $|r_1 \cos(r_2)| \leq 1$, the algorithm performs local search. According to $r_1 = a - at/MaxIt$, when the number of iterations $t > (1 - 1/a)MaxIt$, $r_1 < 1$ and the algorithm no longer performs global search. Therefore, the original r_1 is a monotone decreasing function, which is not good at balancing the global and local search ability of the algorithm. In the middle and late stage, the algorithm is mainly developed locally in a small area, which is easy to fall into the local optimal state. To overcome it, most scholars have studied the modification of it [48]–[51]. Inspired by the waveform change of sine function, we propose a nonlinear amplitude regulating factor r_1^* which is calculated by Eq. (8).

$$r_1^* = a \sin\left(\frac{\pi}{2} \left(\frac{MaxIt + t}{MaxIt}\right)\right) \quad (8)$$

It can be seen from Eq. (8) that r_1^* and the original r_1 are both decreasing functions. However, based on the fluctuation of sine function, r_1^* is beneficial to improve the convergence accuracy of the algorithm for multi-modal and irregular problems while meeting the requirement of large value in the early stage and small value in the late stage of iteration. At the initial stage of iteration, r_1^* decreases slowly, which is beneficial to the population to search for the optimal solution with a large step size and accelerate the convergence rate. At the end of iteration, the rate of r_1^* decline is accelerated, which is conducive to more accurate search of the algorithm in the optimal value neighborhood and improved convergence accuracy. In order to fully prove the effectiveness and superiority of parameter r_1^* , we compared r_1^* with other four expressions of r_1 , as shown in TABLE 1. FIGURE 6 shows a graph of these parameters.

As can be seen from TABLE 1, under the same conditions, when the number of iterations reaches half, $r_1^{(2)} < r_1^{(1)} <$

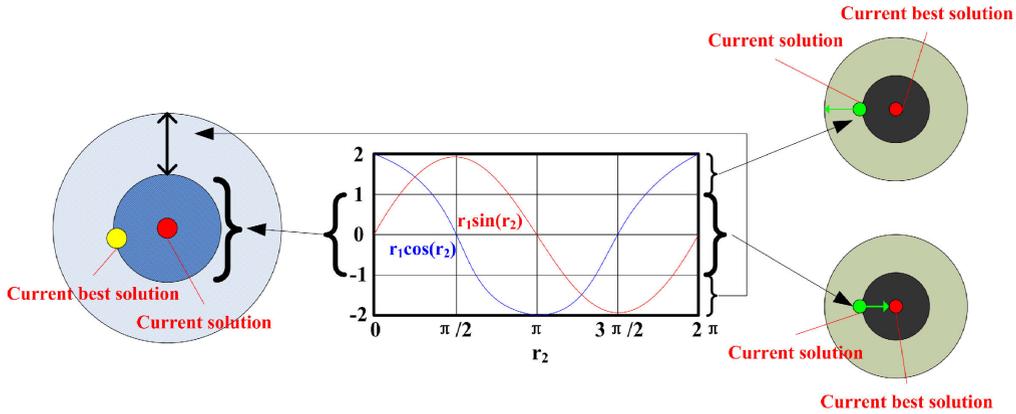


FIGURE 5. When $r_1 = 2, r_2 = 1$, the fluctuation of $r_1 \sin(r_2)$ and $r_1 \cos(r_2)$ and algorithm search strategy.

TABLE 1. Expression comparison of the parameter r_1 .

Expressions for r_1 ($a = 2, b = 0.5$)	when $r_1^{(i)} = 1$	when $t = MaxIt/2$
$r_1^{(2)} = a(1 - t/MaxIt)^2$	$t \approx 0.3096MaxIt$	$r_1^{(2)} \approx 0.5$
$r_1^{(1)} = a(1 - t/MaxIt)$	$t = 0.5MaxIt$	$r_1^{(1)} = 1$
Eq. (8)	$t \approx 0.6667MaxIt$	$r_1^* = \sqrt{2}$
$r_1^{(3)} = ae^{-t/MaxIt}$	$t \approx 0.6931MaxIt$	$r_1^{(3)} \approx 1.2131$
$r_1^{(4)} = a \sin((\pi/2)(1 - t/MaxIt)) + b$	$t \approx 0.8391MaxIt$	$r_1^{(4)} = \sqrt{2} + 0.5$

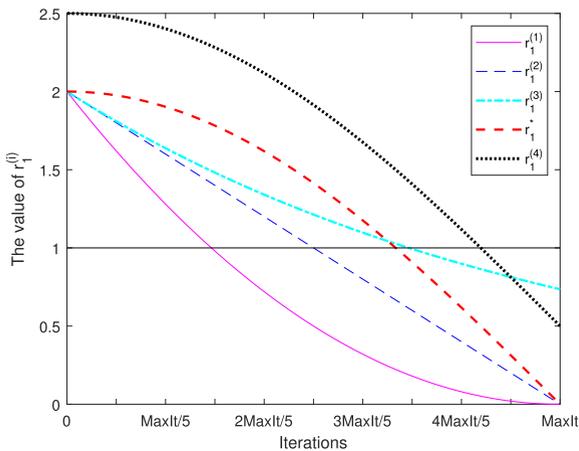


FIGURE 6. Image of parameter $r_1^{(i)}$.

$r_1^{(3)} < r_1^* < r_1^{(4)}$ can be obtained. The global search ability increases successively, while the local development ability weakens. From FIGURE 6, $r_1^{(2)}$ pays more attention to local search, and the algorithm searches for the optimal solution of the problem in a small range most of the time, so it is easy to fall into the local optimal solution. On the contrary, only a small part of values of $r_1^{(3)}$ and $r_1^{(4)}$ are between $[0,1]$, which indicates that the algorithm conducts

global search most of the time. Although the search speed can be improved, the algorithm cannot conduct greatly accurate search in the optimal value neighborhood, thus reducing the convergence accuracy. By contrast, the $r_1^{(*)}$ proposed in this paper can better balance the global search and local development capabilities of the algorithm. While ensuring the convergence speed, the algorithm can conduct more accurate search in a small area.

According to Eq. (7), the original SCA uses individual x_i to guide itself, which has the problem of slow convergence. In addition, x_i may have a low HSI, thus reducing the population quality and affecting the algorithm search. Therefore, this paper uses the elite guidance approach to update the position, so as to speed up the convergence. The specific operation is to replace the guide with the optimal individual x_{best} of the current population, which accelerates the search speed through the x_{best} . Furthermore, in order to make the position information of the current optimal individual x_{best} gradually be fully utilized with the number of iterations, we design a dynamic inertia weight ω inspired by cosine function waveform curve, which makes the algorithm not limited to learning the global optimal value and improves the convergence accuracy. Eq. (9) defines the expression of inertia weight ω .

$$\omega = \gamma \left[1 - \cos \left(\pi \cdot \frac{t}{MaxIt} \right) \right] \quad (9)$$

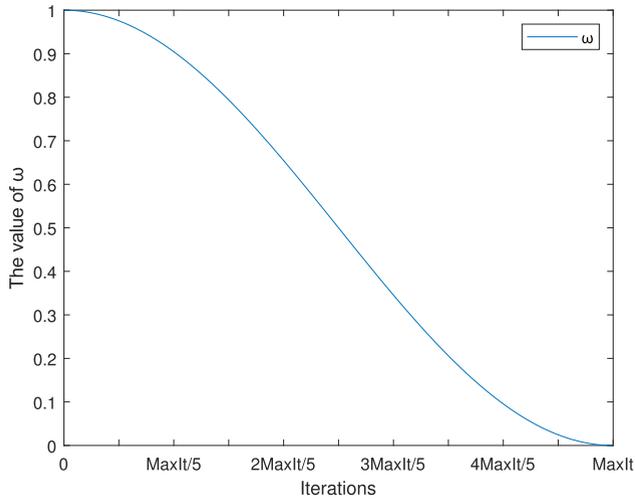


FIGURE 7. The change of inertia weight ω with iterations.

where, γ is the weight adjustment factor and its value is 0.5. FIGURE 7 shows the variation of inertia weight ω with the iterations. In the early stage of evolution, a larger weight is needed to make the particles move to the optimal direction. In the middle of evolution, the inertia weight ω becomes smaller, which preventing the algorithm falling into the local optimal value prematurely, and ensuring the survival and development ability of individuals with low HSI. At the end of evolution, the population gradually does not need elite guidance, so the position update mode gradually degenerates to the unguided update mode with the decrease of the inertia weight ω .

To sum up, for habitat x_i , when a random number generated on a variable is greater than the immigration rate, this variable still needs to be migrated. The new location update mode is implemented through Eq. (10).

$$x_{ij}^{t+1} = \begin{cases} \omega x_{bestj}^t + r_1^* \sin(r_2) |r_3 x_{bestj}^t - x_{ij}^t|, & r_4 < 0.5 \\ \omega x_{bestj}^t + r_1^* \cos(r_2) |r_3 x_{bestj}^t - x_{ij}^t|, & r_4 \geq 0.5. \end{cases} \quad (10)$$

The migration operator of SCBBO is obtained by combining the immigration refusal mechanism, convex migration operator and sine-cosine migration operator. Algorithm 4 presents the migration process of SCBBO.

C. DYNAMIC HYBRID MUTATION OPERATOR

From subsection II-E, BBO can not balance the exploration and exploitation effectively. The standard BBO uses random mutation to generate new individuals, which is weak in generating new solutions with high HSI. Especially in the late stage of evolution, the solution set is close to the theoretical optimal solution, so random mutation is not only difficult to explore better solutions, but also easy to produce more poor solutions. This is also one of the main reasons why BBO cannot effectively solve the high-dimensional optimization problems. In recent years, many scholars have

Algorithm 4 The Migration Operator of SCBBO

```

for  $i = 1$  to  $NP$ 
  for  $j = 1$  to  $D$ 
    if  $rand(0,1) < \lambda_i$ 
      select the  $x_k$  according to the  $\{\mu_k\}_{k=1}^{NP}$ 
      if  $\mu_k > \tau$ 
        update  $x_{ij}$  by Eq. (6)
      else
         $x_{ij} = x_{kj}$ 
      end if
    else
      update  $x_{ij}$  by Eq. (10)
    end if
  end for
end for

```

also improved the BBO mutation operator, but they all have different defects. For instance, MTBBO [52] proposed in 2020 divides the population into three different grades, and carries out different mutations for individuals in different grades. Although it can effectively improve the convergence accuracy, it also increases the computational complexity. The algorithm needs to perform three mutation operators on the population, which consumes a lot of time. Both PRBBO [10] and HGBBO [53] adopted gaussian mutation help BBO to improve population diversity. However, the step size of gaussian distributed random number is short, which can not greatly help the algorithm to escape from the local optimal solution. In addition, NBBO [17], EMBBO [11] and WRBBO [54] directly delete the mutation operator to avoid random mutation generating inferior solutions. Although the computation is reduced, the algorithm only relies on the migration operator to search new solutions, which has the problem of slow convergence speed, and the population diversity decreases rapidly, and the algorithm is easy to fall into the local optimal state. Therefore, this paper proposes a Laplace-Gauss hybrid mutation strategy that dynamically adapts the iterations, which can balance the search and development of the algorithm and help it escape from local extremums.

The probability density functions of Laplacian distribution [55] and Gaussian distribution are defined as Eq. (11) and Eq. (12) respectively.

$$f_L(x; \alpha, \beta) = \frac{1}{2\beta} \exp\left(-\frac{|x - \alpha|}{\beta}\right) \quad (11)$$

$$f_G(x; \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-(x - \mu)^2 / 2\sigma^2\right] \quad (12)$$

where, $\alpha \in (-\infty, \infty)$ and $\beta > 0$ are location parameter and proportion parameter respectively. μ is the mean, σ is the variance. The Laplacian distribution $Lap(\alpha, \beta)$ is defined by the distribution function shown in Eq. (13), which is always

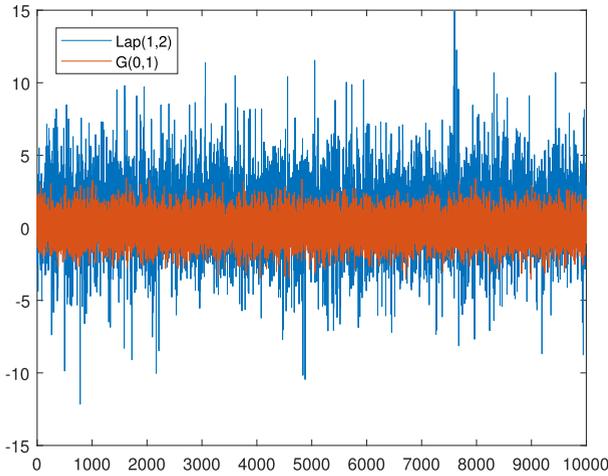


FIGURE 8. Step size comparison of $Lap(1, 2)$ and $G(0, 1)$.

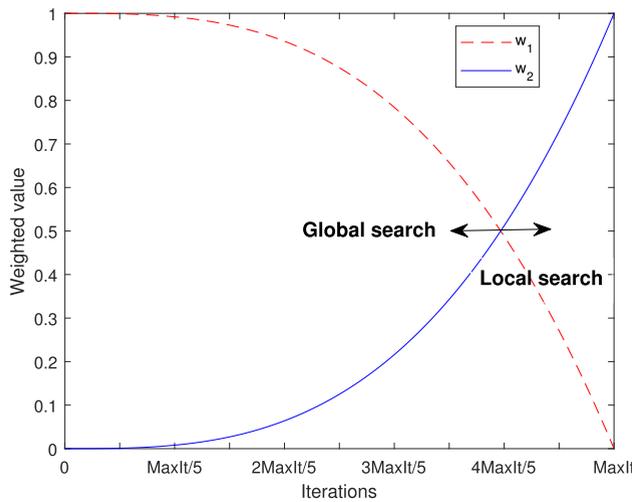


FIGURE 9. The changes of w_1 and w_2 with the iterations.

symmetrically distributed with respect to α .

$$F(x) = \begin{cases} 0.5 \times \exp(-|x - \alpha|/\beta), & x \leq \alpha \\ 1 - 0.5 \times \exp(-|x - \alpha|/\beta), & x > \alpha \end{cases} \quad (13)$$

In order to effectively utilize more random numbers in the search space, $\alpha = 1, \beta = 2$ in the Laplacian distribution $Lap(\alpha, \beta)$ are set. $\mu = 0, \sigma = 1$ in the Gaussian distribution $G(\mu, \sigma)$. Therefore, the mutation formula based on dynamic hybrid strategy is defined as Eq. (14).

$$x_{ij} = w_1 x_{bestj}(1 + Lap(1, 2)) + w_2 x_{bestj}(1 + G(0, 1)) \times \begin{cases} w_1 = 1 - (t/MaxIt)^3 \\ w_2 = (t/MaxIt)^3 \end{cases} \quad (14)$$

Laplacian random number $Lap(1, 2)$ has a larger fluctuation range than Gaussian distribution random number $G(0, 1)$, as shown in FIGURE 8. $Lap(1, 2)$ searches in a larger range of the current optimal value, which is beneficial to maintain the population diversity and help the algorithm escape from the local optimal value. $G(0, 1)$ can search more accurately

within a small range of the current optimal value, which is beneficial to improve the convergence accuracy. Meanwhile, w_1, w_2 are used to adjust the weight between Laplacian random number and Gaussian random number. FIGURE 9 shows the change of weight coefficient with the iterations. w_1 and w_2 are responsible for better exploration and exploitation over the process of iteration. w_1 is used to complete global search, that is exploration; w_2 is used to complete local search, that is exploitation. In Eq. (14), the weight coefficient w_1 of $Lap(1, 2)$ has a large value in the early stage of evolution, so the algorithm can utilize more random numbers and explore better solutions near the current optimal solution with a large step. At the late stage of evolution, the population will converge to the theoretical optimal solution region, and with the increase of the iterations, w_1 gradually decreases, while the weight coefficient w_2 of $G(0, 1)$ keeps increasing. The mutation step size of $G(0, 1)$ is small, which is convenient for the algorithm to search precisely in the optimal solution neighborhood. It not only enhances the local development ability of the algorithm, but also has little influence on the convergence speed in the later period. Therefore, the hybrid mutation strategy avoids falling into the local optimal solution by dynamically adjusting the weight coefficients and improves the search efficiency. Algorithm 5 gives the calculation process of the dynamic hybrid mutation operator.

Algorithm 5 The Mutation Operator of SCBBO

```

for  $i = 1$  to  $NP$ 
  if  $rand(0,1) < m_i$ 
    for  $j = 1$  to  $D$ 
      update  $x_{ij}$  by Eq. (14)
    end for
  end if
end for
    
```

D. DUAL LEARNING STRATEGY

Ergezer et al. [16] proposed the oppositional biogeography-based optimization (OBBO) in 2014, which applied the opposite-based learning strategy into BBO. OBBO merges the reverse individuals of the population into BBO to improve the optimization ability. Opposite-based learning is similar to dual learning, which was first proposed by Collard and Aurand [56]. They designed a genetic algorithm based on dual learning (DGA) to generate dual individuals by taking the maximum Hamming distance of poor individuals in the population. Later, Yang and Yao [57] suggested that only the part of poor individuals in the population should be selected. In this paper, when improving BBO, the dual learning strategy is integrated into the algorithm for the first time.

In SCBBO, when the dual learning strategy is applied to the algorithm, only Nd individuals from the worst part of the population are selected to generate dual individuals. Because it's unlikely that a good individual will produce a better dual individual than the original one. In other words, the individual closer to the optimal value is not worth generating its dual.

Randomly generating dual individuals will not only waste the evaluation times of function, but also reduce the population quality. Therefore, only the dual of the inferior individual will be generated. The specific operation is to take any point x_o on the line segment from W_k to the current optimal individual x_{best} , and then the centrosymmetric point of W_k centered on x_o is the dual individual W'_k . The advantage of this way is to ensure that the dual of the poor individual moves in the direction of the optimal value, and the population evolves in a good direction, thus improving the convergence speed of the algorithm. Algorithm 6 gives the calculation flow of dual learning operator.

Algorithm 6 The Dual Learning Strategy of SCBBO

$\{W_k\} \leftarrow \{\text{The worst } Nd \text{ habitats in the population}\}$
 ($Nd = NP/2$) for each habitat W_k

$r = \text{rand}(0,1)$

$x_o = r \cdot W_k + (1 - r) \cdot x_{best}$

$W'_k = 2 \cdot x_o - W_k$

end for

select the habitats in $\{W_k\} \cup \{W'_k\}$ as the next population

E. GREEDY SELECTION FOR THE BEST SOLUTION

Greedy selection strategy is not the innovation of this paper, but it is essential. Algorithm 7 gives the specific operation. The reason for designing Algorithm 7 is that the optimal individual of the current population is used by the hybrid migration operator, dynamic hybrid mutation operator and dual learning strategy. Only when the optimal individual of each generation does not degenerate can we ensure that the population does not degenerate during evolution. Therefore, in order to achieve the goal without increasing the computation, we only perform greedy selection on the current optimal individual.

Algorithm 7 Greedy Selection for the Best Individual

if $HSI(x_{best}^t) \leq HSI(x_{best}^{t-1})$

$x_{best}^t = x_{best}^{t-1}$

end if

In summary, this paper proposes a dual BBO based on sine cosine algorithm and dynamic hybrid mutation mechanism. SCBBO improves the standard BBO from the contents of the four subsections of this section. Algorithm 8 shows the calculation flow of SCBBO.

IV. CONVERGENCE PROOF OF SCBBO

At present, the convergence of most evolutionary algorithms is proved by Markov model or dynamic model. In this section, we will prove the global convergence of SCBBO with a new method. The establishment of sequence convergence model to prove the convergence of SCBBO is also a major innovation of this paper. For a global optimization problem, assuming that its optimal solution is x^* , then $f(x^*)$ is the

global optimal value. The optimal solution of SCBBO in the t -th iteration is x_{best}^t , and $f(x_{best}^t)$ is the current optimal value. According to the sequence convergence theorem, the equivalent condition that SCBBO can find the global optimal value $f(x^*)$ is that a certain $f(x_{best}^t)$ is in the δ domain of $f(x^*)$, that is, $|f(x_{best}^t) - f(x^*)| \leq \delta$.

During the evolution of SCBBO, each iteration exists a best individual. The set formed by these individuals is:

$$X_{best} = \{x_{best}^1, x_{best}^2, \dots, x_{best}^t, \dots, x_{best}^{MaxIt}\} \quad (15)$$

where, $MaxIt$ is the maximum iteration number. Thus, sequence A can be constructed according to Eq. (16):

$$\begin{cases} A = \{a_1, a_2, \dots, a_{MaxIt}\} \\ a_t = f(x_{best}^t), t = 1, 2, \dots, MaxIt \end{cases} \quad (16)$$

As can be seen from subsection III-E, in SCBBO, the optimal value of each generation will must be better or equivalent that of the previous generation. Therefore, the following formula must be true:

$$a_1 \leq a_2 \leq \dots \leq a_{MaxIt}. \quad (17)$$

With the evolution, the population will gradually move closer to the range where the optimal solution exists, that is, the probability of the optimal individual in the population entering the δ domain of the global optimal solution increases gradually. Eq. (18) is used to express the probability that the optimal value $f(x_{best}^t)$ of the current population converges to the global optimal value $f(x^*)$:

$$p_t = \{|f(x_{best}^t) - f(x^*)| \leq \delta\}, \quad t = 1, 2, \dots, MaxIt \quad (18)$$

According to Eq. (17) and (18), the following relationship must exist:

$$p_1 \leq p_2 \leq \dots \leq p_t, \quad t = 1, 2, \dots, MaxIt \quad (19)$$

therefore, after t -th iteration, the probability that the current optimal value does not converge to the global optimal value is:

$$P_t = (1 - p_1)(1 - p_2) \dots (1 - p_t) \quad (20)$$

From Eq. (19), it can be seen that p_t is monotone and does not decrease, so the following formula is true:

$$\begin{aligned} P_t &= (1 - p_1)(1 - p_2) \dots (1 - p_{t-1})(1 - p_t) \\ &\leq (1 - p_1)(1 - p_1) \dots (1 - p_1)(1 - p_1) \\ &= (1 - p_1)^t \end{aligned} \quad (21)$$

and, since p_1 is the probability, so $0 \leq p_1 \leq 1$, then we have $0 \leq (1 - p_1) \leq 1$. After many iterations, Eq. (22) can be set up.

$$\lim_{t \rightarrow \infty} (1 - p_1)^t = 0 \quad (22)$$

According to Eq. (22), after a large number of iterations, the probability that the algorithm does not converge to the optimal value is 0. Therefore, as the iteration number t

Algorithm 8 The Calculation Procedure of SCBBO

```

initialize SCBBO parameters, which involve  $S_{max}$ ,  $I$ ,  $E$ ,  $NP$ ,  $m_{max}$ ,  $\tau$  and  $Nd$ 
initialize the population of SCBBO by latin hypercube sampling
for each habitat, calculate the HSI and sort from best to worst
for each habitat, calculate the  $S_i$  by Eq. (2), the  $\lambda_i$  and  $\mu_i$  by Eq. (3)
for each habitat, calculate the  $P_i$  by Eq. (4), the  $m_i$  by Eq. (5)
while (the termination condition is not met)
    perform the hybrid migration operator by Algorithm 4
    perform the dynamic hybrid mutation operator by Algorithm 5
    perform the dual learning strategy by Algorithm 6
    perform the greedy selection by Algorithm 7
    for each habitat, calculate the HSI and sort from best to worst
end while
output the optimal solution

```

increases, SCBBO will eventually converge to the global optimal value $f(x^*)$ in the form of probability 1. The proof is completed.

V. SCBBO COMPLEXITY DISCUSS

With reference to BBO, the computational complexity of SCBBO is compared and analyzed. As for the calculation process of the algorithms, comparing Algorithm 3 and Algorithm 8, it can be seen that SCBBO moves the calculation of habitat immigration rate, emigration rate, species probability and mutation rate out of the iteration loop. Because they are all based on rankings, so there is no need to double count. However, the original BBO does not avoid repeated calculation. In each iteration, it recalculates the immigration rate, emigration rate, species probability and mutation rate of each habitat, and the total calculation times is $4 \cdot MaxIt \cdot NP$. On the contrary, SCBBO only calculates these data once in the whole evolutionary process, so the total number of calculations is $4 \cdot NP$. Therefore, SCBBO greatly reduces the computational complexity of migration operator and mutation operator, and saves the calculation of $4 \cdot (MaxIt - 1) \cdot NP$ times at least. Although the sine-cosine migration operator of SCBBO adds some judgment steps, it does not bring additional loops. Finally, the dual learning strategy is added to SCBBO. In each iteration, Nd individuals with low HSI will be selected to generate their dual individuals. For every dual individual generated, two additional calculations are performed. So the number of computations increases to $2 \cdot Nd \leq 2 \cdot NP$ in each generation, and the total number of computations does not exceed $2 \cdot MaxIt \cdot NP$. Although SCBBO increases the computation in the dual learning strategy, this has been compensated in the migration and mutation operators.

To sum up, SCBBO fully reduces the computation amount from migration and mutation operators, thus achieving the purpose of reducing the running time. Although the process of generating dual individuals increases some computational complexity, it is fully compensated in the process of migration and mutation. SCBBO as a whole saves at least $(2 \cdot MaxIt - 4) \cdot NP$ calculations compared with BBO.

VI. EXPERIMENT AND ANALYSIS

In order to verify the effectiveness and superiority of SCBBO on solving high-dimensional global optimization problems, we carry out a series of simulation experiments. As shown in TABLE 2, this paper summarizes and selects 24 classical benchmark functions. They contain complex functions such as unimodal, multimodal, irregular, rotating and noisy, which can test the comprehensive ability of the algorithms. The experimental environment for all numerical experiments in this paper is MATLAB 2020a.

A. COMPARISON BETWEEN SCBBO AND TWO CONSTITUENT ALGORITHMS

This subsection compares SCBBO with its two constituent algorithms to verify that SCBBO improves the performance of BBO and SCA. SCBBO does not increase additional function evaluation numbers compared with BBO. So the same iterations number means the same function evaluation number. Therefore, we set $MaxIt = 1000$, and the maximum mutation rate m_{max} of BBO is 0.05 [58]. To avoid contingency, each algorithm runs 50 times independently on 24 benchmark functions on 30, 50, and 100 dimensions, respectively. Finally, the mean (Mean) and standard deviation (Std) of the 50 errors are calculated, and the rank-sum test will be performed at a significance level of 0.05. TABLE 3 show the rank-sum test results, and the last line summarizes the result of this comparison. The test results are obtained as “(w/t/l)”, whose representative meaning is: w(+:win)/t(≈:tie)/l(-:lose). Where, “-” means that the performance of the contestant algorithm is worse than that of SCBBO, “+” means that the performance of the contestant algorithm is superior, and “≈” means that the performance of contestant algorithm and SCBBO is similar.

As can be seen from TABLE 3, the overall performance of SCBBO is significantly better than BBO and SCA on 30, 50 and 100 dimensions, so the performance of two original algorithms are greatly improved. BBO performs worse than SCBBO on all benchmark functions. SCA and has

TABLE 2. 24 Benchmark functions.

Function	Search Space	$f(x^*)$
$f1(x) = \sum_{i=1}^D ix_i^2$	$[-10, 10]^D$	0
$f2(x) = \sum_{i=1}^D x_i + \prod_{i=1}^D x_i $	$[-10, 10]^D$	0
$f3(x) = \sum_{i=1}^D x_i^2 + \left(\sum_{i=1}^D 0.5ix_i\right)^2 + \left(\sum_{i=1}^D 0.5ix_i\right)^4$	$[-5, 10]^D$	0
$f4(x) = \max_{i=1}^D \{ x_i \}$	$[-100, 100]^D$	0
$f5(x) = \sum_{i=1}^D \left(\sum_{j=1}^i x_j\right)^2 \times (1 + 0.4 N(0, 1))$	$[-100, 100]^D$	0
$f6(x) = \sum_{i=1}^D \left(\sum_{j=1}^i x_j\right)^2$	$[-100, 100]^D$	0
$f7(x) = \sum_{i=1}^D ix_i^4 + rand$	$[-100, 100]^D$	0
$f8(x) = \sum_{i=1}^D z_i^2 - 450, z = x - o$	$[-100, 100]^D$	-450
$f9(x) = \sum_{i=1}^D x_i ^{i+1}$	$[-1, 1]^D$	0
$f10(x) = \exp\left(0.5 \sum_{i=1}^D x_i \right) - 1$	$[-1.28, 1.28]^D$	0
$f11(x) = \sum_{i=1}^D (10^6)^{\frac{i-1}{D-1}} x_i^2$	$[-100, 100]^D$	0
$f12(x) = \sum_{i=1}^D [x_i + 0.5]^2$	$[-100, 100]^D$	0
$f13(x) = \sum_{i=1}^D [z_i^2 - 10 \cos(2\pi z_i) + 10] - 330, z = x - o$	$[-5.12, 5.12]^D$	-330
$f14(x) = \sum_{i=1}^D [z_i^2 - 10 \cos(2\pi z_i) + 10], z_i = \begin{cases} x_i, x_i < 0.5 \\ \text{round}(2x_i)/2, \text{ else} \end{cases}$	$[-5.12, 5.12]^D$	0
$f15(x) = -20 \exp\left(-0.2\sqrt{\sum_{i=1}^D \frac{z_i^2}{D}}\right) - \exp\left[\frac{1}{D} \sum_{i=1}^D \cos(2\pi z_i)\right] + e - 120, z = x - o$	$[-32, 32]^D$	-140
$f16(x) = \sum_{i=1}^D x_i \sin(x_i) + 0.1x_i $	$[-10, 10]^D$	0
$f17(x) = \frac{1}{4000} \left[\sum_{i=1}^D (z_i - 100)^2\right] - \left[\prod_{i=1}^D \cos\left(\frac{z_i - 100}{\sqrt{i}}\right)\right] - 179, z = x - o$	$[-600, 600]^D$	-180
$f18(x) = -\cos\left(2\pi\sqrt{\sum_{i=1}^D x_i^2}\right) + 0.1 \times \sqrt{\sum_{i=1}^D x_i^2} + 1$	$[-100, 100]^D$	0
$f19(x) = \sum_{i=1}^D \left\{ \sum_{k=0}^{k_{\max}} [a^k \cos(2\pi b^k (x_i + 0.5))] \right\} - D \sum_{k=0}^{k_{\max}} [a^k \cos(2\pi b^k \times 0.5)],$ $a = 0.5, b = 3, k_{\max} = 20$	$[-0.5, 0.5]^D$	0
$f20(x) = \frac{\pi}{D} \{10 \sin^2(\pi y_i) + \sum_{i=1}^{D-1} (y_i - 1)^2 [1 + 10 \sin^2(\pi y_{i+1})] + (y_D - 1)^2\} + \sum_{i=1}^D u(x_i, 10, 100, 4)$ $y_i = 0.25(x_i + 1) + 1, u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m, x_i > a \\ 0, -a \leq x_i \leq a \\ k(-x_i - a)^m, x_i < -a \end{cases}$	$[-50, 50]^D$	0
$f21(x) = 0.1\{\sin^2(3\pi x_1) + \sum_{i=1}^{D-1} (x_i - 1)^2 [1 + \sin^2(3\pi x_{i+1})] + (x_D - 1) [1 + \sin^2(2\pi x_D)]\} + \sum_{i=1}^D u(x_i, 5, 100, 4)$	$[-50, 50]^D$	0
$f22(x) = \sum_{i=1}^{n-1} F(x_i, x_{i+1}) + F(x_n, x_1), F(x, y) = (x^2 + y^2)^{0.25} \cdot [\sin^2(50(x^2 + y^2)^{0.1}) + 1]$	$[-100, 100]^D$	0
$f23(x) = \sum_{i=1}^{D-1} [x_i^2 + 2x_{i+1}^2 - 0.3 \cos(3\pi x_i) \cos(3\pi x_{i+1}) + 0.3]$	$[-100, 100]^D$	0
$f24(x) = \sum_{i=1}^{D-1} [100(z_{i+1} - z_i^2)^2 + (z_i - 1)^2] + 390, z = x - o + 1$	$[-100, 100]^D$	390

similar performance to SCBBO on two benchmark functions ($f9$ with $D = 30$, $f10$ with $D = 30$ and 50), but has a bigger error mean on the remaining 22 functions. Horizontal comparison: SCBBO's convergence accuracy is much higher than BBO and SCA. It can be seen that the improvement strategies in this paper effectively improves the search ability of the algorithm. The hybrid migration operator makes the population close to the optimal solution quickly, and the dynamic mutation operator keeps the diversity of the population and balances the exploration and balance of the algorithm effectively. Longitudinal comparison: The convergence accuracy of SCBBO basically does not decrease with the increase of the dimensions. This shows that the improvement strategies in this paper makes BBO suitable for high-dimensional optimization environment and has good scalability. This is because the Latin hypercube sampling method makes SCBBO populations on different dimensions have good ergodicity, while the dual learning strategy can make the poor individuals in the population close to the optimal solution. At the same time, the hybrid migration operator and dynamic mutation operator enable SCBBO to switch freely between global search and

local search, so that it is not easy to fall into the local optima, and can still converge quickly in high dimensional environment.

Next, we compare the three algorithms' convergence speed and stability. According to the 50 times optimization results of the three algorithms on 24 benchmark functions on 30, 50 and 100 dimensions, boxplots are drawn to compare the stability. At the same time, a search with the smallest error and the errors of each generation are selected to make the best convergence graphs of three algorithms to compare the convergence. As shown in FIGURE 3.

From FIGURE 3, SCBBO converges faster than BBO and SCA on different benchmark functions. Although SCA is similar to that of SCBBO on some benchmark functions, the convergence curve of SCA fluctuates greatly (e.g. $f7, f9, f14, f17, f18$ and $f24$) and the algorithm is unstable. On functions $f7, f9, f11$ and $f24$, SCA does not converge. This shows that SCA is easy to fall into the local optimal solutions. On the contrary, the convergence curve of SCBBO basically does not fluctuate, and the algorithm can successfully escape from the local optima. This is because the hybrid migration operator accelerates the convergence

TABLE 3. Rank-sum test results of SCBBO and its two component algorithms with $D = 30, 50$ and 100 .

F	D	BBO		SCA		SCBBO	
		Mean	Std	Mean	Std	Mean	Std
f_1	$D = 30$	1.59E+00	5.98E-01	2.21E-03	1.20E-04	0.00E+00	0.00E+00
	$D = 50$	1.56E+01	2.18E+01	3.97E-04	2.38E-06	0.00E+00	0.00E+00
	$D = 100$	3.59E+02	2.81E+03	6.24E-02	9.13E-02	0.00E+00	0.00E+00
f_2	$D = 30$	1.24E+00	4.14E-02	6.36E-12	2.02E-21	0.00E+00	0.00E+00
	$D = 50$	3.65E+00	2.39E-01	1.23E-13	3.68E-25	0.00E+00	0.00E+00
	$D = 100$	1.61E+01	1.23E+00	2.03E-11	9.24E-21	0.00E+00	0.00E+00
f_3	$D = 30$	1.17E+01	2.16E+01	5.97E-03	1.60E-03	0.00E+00	0.00E+00
	$D = 50$	6.46E+01	4.59E+02	7.37E-01	1.33E+01	0.00E+00	0.00E+00
	$D = 100$	8.06E+02	2.47E+04	8.67E-03	1.76E-03	0.00E+00	0.00E+00
f_4	$D = 30$	1.02E+01	3.80E+00	6.07E+00	7.14E+01	0.00E+00	0.00E+00
	$D = 50$	2.01E+01	3.78E+00	4.11E+00	4.12E+01	0.00E+00	0.00E+00
	$D = 100$	4.12E+01	7.44E+00	8.92E-01	3.15E+00	0.00E+00	0.00E+00
f_5	$D = 30$	2.23E+06	2.39E+12	2.07E+03	8.65E+07	0.00E+00	0.00E+00
	$D = 50$	5.88E+08	1.12E+17	4.57E+04	5.12E+10	0.00E+00	0.00E+00
	$D = 100$	9.37E+11	1.73E+23	8.81E+04	1.86E+11	0.00E+00	0.00E+00
f_6	$D = 30$	1.96E+03	7.96E+05	1.38E+00	1.06E+01	0.00E+00	0.00E+00
	$D = 50$	2.38E+04	6.11E+07	2.65E+01	5.32E+03	0.00E+00	0.00E+00
	$D = 100$	9.67E+05	4.60E+10	1.42E+04	4.89E+09	0.00E+00	0.00E+00
f_7	$D = 30$	5.60E-02	3.33E-04	5.87E-02	4.70E-03	2.83E-05	5.31E-10
	$D = 50$	1.89E-01	3.06E-03	5.35E-02	2.42E-03	2.87E-05	6.43E-10
	$D = 100$	1.12E+00	5.25E-02	5.32E-02	1.07E-02	3.19E-05	1.04E-09
f_8	$D = 30$	1.38E+01	2.55E+01	3.33E-02	8.28E-03	0.00E+00	0.00E+00
	$D = 50$	6.72E+01	3.60E+02	1.10E-09	2.95E-17	0.00E+00	0.00E+00
	$D = 100$	8.24E+02	1.41E+04	2.14E-02	1.12E-02	0.00E+00	0.00E+00
f_9	$D = 30$	1.31E-05	1.45E-09	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	$D = 50$	1.88E-05	7.84E-10	1.13E-07	1.27E-13	0.00E+00	0.00E+00
	$D = 100$	1.85E-05	1.01E-09	4.00E-06	4.88E-11	0.00E+00	0.00E+00
f_{10}	$D = 30$	9.08E-02	2.29E-04	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	$D = 50$	2.74E-01	1.23E-03	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	$D = 100$	2.30E+00	2.02E-01	1.07E-15	1.34E-29	0.00E+00	0.00E+00
f_{11}	$D = 30$	3.96E+05	8.09E+10	4.54E+01	5.03E+04	0.00E+00	0.00E+00
	$D = 50$	1.33E+06	7.60E+11	4.63E+03	1.43E+08	0.00E+00	0.00E+00
	$D = 100$	7.86E+06	8.20E+12	3.31E+05	1.91E+12	0.00E+00	0.00E+00
f_{12}	$D = 30$	1.47E+01	1.83E+01	8.00E-02	1.57E-01	0.00E+00	0.00E+00
	$D = 50$	7.02E+01	2.84E+02	1.12E+00	1.60E+01	0.00E+00	0.00E+00
	$D = 100$	8.30E+02	1.55E+04	5.20E+00	3.04E+01	0.00E+00	0.00E+00
$w/t/l$	$D = 30$	0/0/12		0/0/12			
	$D = 50$	0/0/12		0/0/12			
	$D = 100$	0/0/12		0/0/12			

speed of SCBBO, while the dynamic mutation operator helps it escape from the local optima. At the same time, greedy selection ensures that the population does not degenerate, so the convergence curve of SCBBO always decreases. This also verifies the convergence proof in Section IV. At the same time, the 50 error values obtained by SCBBO on different benchmark functions are very close, and the boxplot can hardly be seen. However, BBO and SCA both have multiple outliers, and the search performance is unstable. Therefore, by carefully observing the convergence curves and boxplots, it can be seen that SCBBO has better convergence and robustness than BBO and SCA, which verify the effectiveness of the improvement strategies again.

B. COMPARISON BETWEEN SCBBO AND OTHER BBO VARIANTS

This section compares the SCBBO with seven excellent BBO variants. They are all proposed in the past five

years, and TABLE 4 shows the detailed information of them. Consistent with subsection VI-A, eight algorithms search the optimal values on 24 benchmark functions. The performance of these BBO variants on low-dimensional optimization problems is fully verified in their original reference articles. So we mainly compare the performance of SCBBO and them on high dimensions. In literatures [48], [61]–[63], the dimensions greater than 100 are defined as high-dimensional optimization problems. Thus, SCBBO and the seven BBO variants search the optimal values of 24 benchmark functions on 200 dimensions. Set ($MFEs$) of each function, and record the error between the optimal value searched and the theoretical optimal value when reaching $MFEs$. According to the method of setting $MFEs$ in CEC2017 [64], the $MFEs$ of each benchmark function is: $\text{dimension}(D) \times 10^4$.

Similarly, in order to avoid contingency, each algorithm independently runs 50 times, then the mean and standard

TABLE 3. (Continued.) Rank-sum test results of SCBBO and its two component algorithms with $D = 30, 50$ and 100 .

F	D	BBO		SCA		SCBBO	
		Mean	Std	Mean	Std	Mean	Std
f_{13}	$D = 30$	4.81E+00	2.16E+00	4.95E-04	3.77E-06	0.00E+00	0.00E+00
	$D = 50$	1.69E+01	9.46E+00	2.12E-12	7.35E-23	0.00E+00	0.00E+00
	$D = 100$	8.60E+01	1.08E+02	4.91E-08	5.87E-14	0.00E+00	0.00E+00
f_{14}	$D = 30$	4.83E+00	1.95E+00	8.30E-02	1.69E-01	0.00E+00	0.00E+00
	$D = 50$	1.49E+01	6.94E+00	2.16E-08	1.12E-14	0.00E+00	0.00E+00
	$D = 100$	6.16E+01	1.86E+01	4.97E-03	6.05E-04	0.00E+00	0.00E+00
f_{15}	$D = 30$	1.87E+00	1.10E-01	7.14E-01	2.45E+00	8.88E-16	0.00E+00
	$D = 50$	2.82E+00	5.26E-02	5.97E+00	1.20E+01	8.88E-16	0.00E+00
	$D = 100$	4.98E+00	8.96E-02	7.53E+00	1.49E+01	8.88E-16	0.00E+00
f_{16}	$D = 30$	8.61E-02	6.44E-04	1.11E-04	2.93E-07	0.00E+00	0.00E+00
	$D = 50$	3.74E-01	7.45E-03	2.58E-07	1.64E-12	0.00E+00	0.00E+00
	$D = 100$	3.56E+00	2.04E-01	3.00E-07	2.20E-12	0.00E+00	0.00E+00
f_{17}	$D = 30$	1.11E+00	1.37E-03	1.07E-02	5.72E-02	0.00E+00	0.00E+00
	$D = 50$	1.57E+00	3.10E-02	3.11E-02	2.37E-02	0.00E+00	0.00E+00
	$D = 100$	8.88E+00	2.06E+00	1.18E-02	2.57E-03	0.00E+00	0.00E+00
f_{18}	$D = 30$	2.09E+00	8.31E-02	1.28E-03	2.54E-04	0.00E+00	0.00E+00
	$D = 50$	3.77E+00	1.95E-01	6.10E-02	2.86E-03	0.00E+00	0.00E+00
	$D = 100$	8.87E+00	3.99E-01	4.10E-01	4.10E-02	0.00E+00	0.00E+00
f_{19}	$D = 30$	3.20E+00	1.24E-01	5.68E-16	7.91E-30	0.00E+00	0.00E+00
	$D = 50$	7.47E+00	4.05E-01	7.21E-11	2.17E-19	0.00E+00	0.00E+00
	$D = 100$	2.59E+01	2.16E+00	1.31E-10	3.91E-19	0.00E+00	0.00E+00
f_{20}	$D = 30$	9.49E-02	1.35E-03	1.77E+00	1.01E+00	2.19E-02	7.99E-04
	$D = 50$	3.01E-01	1.50E-02	1.50E+00	1.66E-01	2.29E-01	2.74E-03
	$D = 100$	3.34E+04	1.85E+10	2.54E+04	1.30E+10	4.34E-01	3.29E-03
f_{21}	$D = 30$	1.62E+00	4.68E-02	3.61E+02	2.65E+06	2.10E-01	6.30E-03
	$D = 50$	3.66E+00	1.82E+00	4.81E+03	5.64E+08	3.17E+00	2.15E-01
	$D = 100$	3.86E+05	3.18E+11	3.60E+01	1.08E+04	8.62E+00	1.27E-01
f_{22}	$D = 30$	4.11E+01	2.87E+01	5.25E-13	2.71E-24	0.00E+00	0.00E+00
	$D = 50$	8.82E+01	6.35E+01	9.87E-12	1.40E-21	0.00E+00	0.00E+00
	$D = 100$	2.65E+02	2.00E+02	1.48E-10	4.74E-19	0.00E+00	0.00E+00
f_{23}	$D = 30$	4.56E+01	2.15E+02	4.40E-01	3.69E+00	0.00E+00	0.00E+00
	$D = 50$	1.92E+02	2.53E+03	4.59E-01	2.40E+00	0.00E+00	0.00E+00
	$D = 100$	2.49E+03	1.90E+05	1.07E+00	2.24E+01	0.00E+00	0.00E+00
f_{24}	$D = 30$	8.95E+03	4.79E+07	1.12E+05	1.58E+11	2.68E+01	7.55E-01
	$D = 50$	8.41E+04	2.49E+09	2.77E+05	8.16E+11	4.80E+01	7.29E-01
	$D = 100$	5.63E+06	5.73E+12	8.99E+05	1.72E+09	9.82E+01	4.79E-01
$w/t/l$	$D = 30$	0/0/12		0/0/12			
	$D = 50$	0/0/12		0/0/12			
	$D = 100$	0/0/12		0/0/12			

deviation of 50 errors are used as evaluation indexes. The mean reflects the searching ability and the standard deviation reflects the stability of the algorithm. Therefore, the mean is the focus of comparison. TABLE 5 shows the optimization results of SCBBO and seven BBO variants for 24 benchmark functions on 200 dimensions. Where, the boldface represents the best result of the eight algorithms, and the last column represents the rank-sum test results, which has the same meaning as that in TABLE 3.

As can be seen from TABLE 5, SCBBO obviously has the best overall performance among all the BBO variants. PRBBO gets the same result as SCBBO on f_{12} and worse on all remaining functions. Then, the error mean of BBOSB is better that of SCBBO on only one benchmark function (f_{21}), and the results are not as good as SCBBO on the remaining 23 functions. While, TDBBO, WRBBO, DCGBBO and FABBO have larger mean and standard deviation than SCBBO on all benchmark functions.

In contrast, HGBBO is more competitive. It has better mean and standard deviation than SCBBO on three benchmark functions (f_{20}, f_{21} and f_{24}), and the same result as SCBBO on f_{12} . It can be seen that although these BBO variants have excellent performance in low-dimensional environment, they cannot effectively solve high-dimensional global optimization problems. They are not suitable for high-dimensional environment. Conversely, even when $D = 200$, SCBBO still converges precisely to the theoretical optimal values on 19 benchmark functions ($f_1 - f_6, f_8 - f_{14}, f_{16} - f_{19}, f_{22}$ and f_{23}). This shows that SCBBO has good scalability and is suitable for high-dimensional optimization environment. Latin hypercube sampling method makes the initial population of SCBBO still have good ergodicity in high dimensional space. Hybrid migration operator enables SCBBO to search effectively and speeds up convergence. Then, dynamic hybrid mutation operator makes SCBBO not easy to fall into the local optima. Dual learning strategy helps

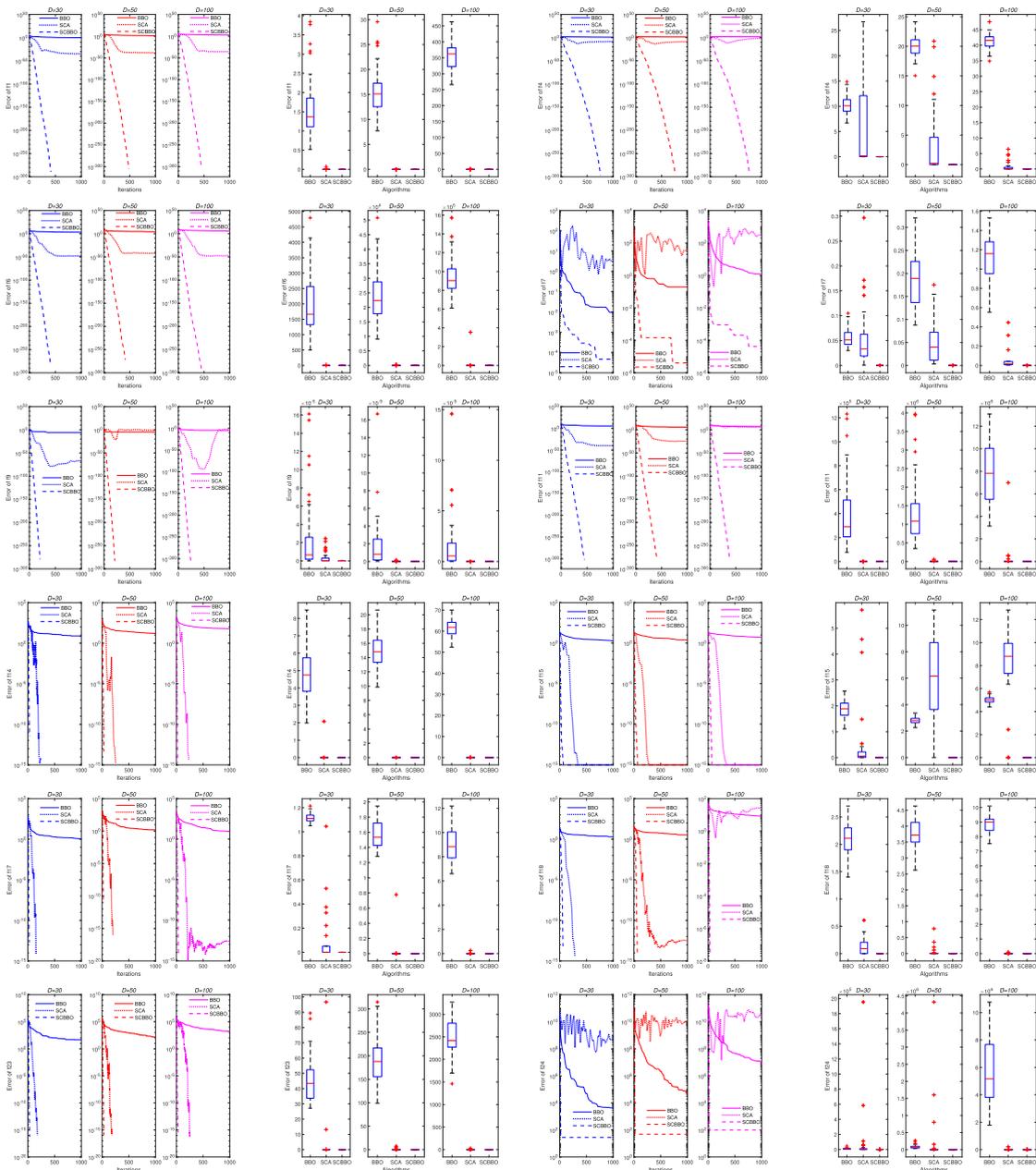


FIGURE 10. Convergence curves and boxplots of BBO, SCA and SCBBO on different benchmark functions with $D = 30, 50$ and 100 .

the poor individuals of the population move quickly to the optimal solution.

Similarly, to clearly compare the convergence process between SCBBO and the seven BBO variants, we plot the convergence curves and boxplots of them on different benchmark functions. As shown in FIGURE 11, SCBBO has the fastest convergence speed on all functions, and it is not easy to fall into the local optima. On $f1, f2, f4, f6, f11$ and $f22$, SCBBO's convergence speed is faster than other algorithms at the beginning of iteration. When the iteration enters the middle, SCBBO rapidly converges to the optimal value, and the number of iterations is at least 1000 times less than other algorithms. Especially for functions $f13, f14, f15, f18$ and

$f19$, the convergence curves of SCBBO are almost invisible. From boxplots, SCBBO has excellent robustness and stable search performance in the face of different high-dimensional benchmark functions.

In summary, the overall performance of SCBBO is better than that of PRBBO, BBOSB, TDBBO, WRBBO, HGBBO, DCGBBO and FABBO. SCBBO has higher convergence accuracy, faster convergence speed and better stability on high-dimensional global optimization problems.

C. COMPARISON BETWEEN SCBBO AND OTHER EAs

To further verify the superiority of SCBBO for solving high-dimensional global optimization problems, we compare

TABLE 4. Details of seven BBO variants.

Year	Strategy	Name	Parameter Setting
2017	BBO+Random ring topology mixed migration +Adaptive Powell method	PRBBO [10]	$I = E = 1, m_{\max} = 0.005, \varepsilon = 0.1, limit = 100$
2018	BBO+Brain storm optimization (BSO)+Greedy select	BBOSB [59]	$I = E = 1, m_{max} = 0.05, perc_e = 20$ $P_{rep} = 0.2, P_e = 0.6, P_{one} = 0.8, K = 25$
2019	BBO+Two-stage differential mechanism +Gaussian mutation	TDBBO [46]	$I = E = 1, m_{max} = 0.01, \mu = 0, \sigma = 1, c = 0.3$
2019	BBO+Random scale differential mutation +Dynamic heuristic crossover+Elite storage	WRBBO [54]	$I = E = 1, \alpha_d = rand^\beta, \beta = \sqrt{t/MaxIt}$
2020	BBO+Hybrid migration+Global-best Gaussian mutation+Random opposite-based learning	HGBBO [53]	$I = E = 1, mp_{max} = 0.02, mp_{min} = 0.001, \beta_{max} = 13$
2021	BBO+Dynamic cross migration+dynamic Gaussian mutation+Trigger ablation rule hierarchical tissue P-system	DCGBBO [60]	$I = E = 1, m_{max} = 0.01, K = 10, a = 0.02$
2021	BBO+Firefly algorithm (FA)	FABBO [30]	$I = E = 1, \gamma = 0.1, K_x = 0, K_y = 1$

it with seven state-of-the-art EAs proposed in the past few years: GWO [27], WOA [65], SSA [66], AEFA [67], AOA [68], ChOA [69], MPA [70]. Among them, GWO, WOA, SSA and AEFA are highly cited advanced algorithms in “Web of Science”, which are called state-of-the-art algorithms. AOA, ChOA and MPA are the novel algorithms proposed in the past two years, which are suggested to solve high-dimensional global optimization problems. Therefore, SCBBO can further verify its superiority by comparing with these outstanding algorithms.

Then, eight algorithms search for optimization on 24 benchmark functions on 500 dimensions, and record the optimal value searched when $MFEs$ is reached. Similarly, in order to avoid contingency, each algorithm independently runs 50 times, and the mean and standard deviation of 50 times errors are used as evaluation indexes. TABLE 6 shows the rank-sum test results of eight algorithms, in which the best results are shown in bold.

From TABLE 6, SCBBO still has the best overall performance among the eight evolutionary algorithms, with the minimum mean value obtained on 21 functions and zero error on 20 functions. WOA converges to the theoretical optimal value on eight benchmark functions ($f8, f9, f12 - f14, f17$ and $f19$), which is consistent with the results of SCBBO, while the results on the remaining functions are inferior to SCBBO. ChOA converges to the theoretical optimal value only on $f12$, but the mean and standard deviation of the remaining 23 functions are greater than SCBBO. Then, GWO gets better results than SCBBO on $f20$, and the same results on the other five functions ($f12 - f14, f23$ and $f24$), but worse results on the remaining 18 functions. However, the mean and standard deviation of all functions obtained by AEFA and AOA are inferior to SCBBO. In contrast, SSA and MPA are more competitive. The results obtained by SSA on three functions ($f20, f21$ and $f24$) are the best among the eight algorithms. The results of MPA are better than SCBBO on three functions ($f20, f21$ and $f24$) and

identical to SCBBO on eight functions ($f9, f10, f12 - f14, f17, f19$ and $f23$). So, although these advanced algorithms show excellent performance on low-dimensional problems, their performance significantly decreases when solving high-dimensional optimization problems. On the contrary, even when $D = 500$, SCBBO converges precisely to the theoretical optimal value on 19 functions. A careful comparison between TABLE 5 and TABLE 6 shows that the convergence results of SCBBO on 200 dimensions are almost the same as on 500 dimensions. Therefore, the improvement strategies in this paper makes BBO suitable for high-dimensional optimization environment, and the algorithm performance has good ductility.

For a better evaluation of SCBBO and the compared algorithms, FIGURE 12 shows convergence curves of the eight algorithms on different functions. It can be observed that SCBBO algorithm converges much faster than other EAs on different benchmark functions, saving at least 800 iterations and not falling into the local optima. Especially for functions $f15, f18$ and $f23$, SCBBO converges rapidly and the convergence curve is almost invisible. Then, the convergence curve of WOA on $f7$ is unstable, while SCBBO maintains a smooth convergence curve. This is because the greedy selection ensures that the population does not degenerate, but always converges towards the optimal solution. Therefore, even compared with the advanced evolutionary algorithms, the algorithm proposed in this paper also shows the optimal search performance and stability.

In a word, on high-dimensional global optimization problems, the performance of SCBBO is significantly better than that of GWO, WOA, SSA, AEFA, AOA, ChOA and MPA in both solution quality and convergence speed.

D. PERFORMANCE COMPARISON OF SCBBO ON DIFFERENT HIGH DIMENSIONS

With the rapid development of the present society, the practical problems in life have higher and higher requirements

TABLE 5. Rank-sum test results of SCBBO and BBO variants with $D = 200$.

F	PRBBO		BBOSS		TDBBO		WRBBO		HGBBO		DCCBBO		FABBO		SCBBO	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std
f_1	1.26E-02	2.16E-05	2.01E+03	7.58E+04	5.45E+03	5.00E+03	2.15E+03	1.39E+07	1.64E-15	1.56E-30	2.09E+03	1.13E+07	1.28E-13	7.13E-28	0.00E+00	0.00E+00
f_2	2.97E-02	4.59E-05	3.35E+01	3.51E+00	1.44E+02	6.73E+02	4.53E+01	4.68E+03	3.58E-11	1.16E-22	4.69E+01	1.35E+03	3.10E-07	6.71E-16	0.00E+00	0.00E+00
f_3	1.71E-02	3.68E-05	2.15E+01	3.51E+00	6.69E+03	1.24E+06	3.32E+03	4.05E+07	6.43E-17	8.50E-34	2.71E+03	2.40E+07	1.07E-13	3.31E-28	0.00E+00	0.00E+00
f_4	2.99E+01	1.79E+01	2.88E+01	2.17E+00	7.16E+01	7.83E+01	9.34E+01	1.40E+02	1.63E-02	1.63E-05	9.48E+01	8.46E+00	7.08E+01	2.87E+00	0.00E+00	0.00E+00
f_5	2.11E+09	1.46E+18	3.72E+12	2.20E+24	8.42E+14	1.01E+29	8.17E+15	6.08E+32	1.54E-14	1.43E-28	4.51E+16	1.25E+33	2.87E+11	9.90E+22	0.00E+00	0.00E+00
f_6	5.01E+01	2.44E+02	3.27E+05	4.65E+09	7.07E+07	4.62E+14	2.35E+07	2.22E+15	5.61E-12	1.93E-23	2.35E+07	8.49E+14	2.35E-09	7.49E-19	0.00E+00	0.00E+00
f_7	2.26E-01	1.66E-03	1.28E+02	2.38E+02	6.89E+01	1.10E+02	3.18E-01	2.40E-02	1.59E-02	1.85E-05	1.74E+00	2.99E-01	2.11E-01	8.34E-03	2.72E-05	7.73E-10
f_8	1.35E-02	1.74E-05	2.16E+01	6.28E+00	7.18E+03	1.43E+06	3.83E+03	2.79E+07	6.43E-17	7.43E-34	1.50E+03	1.26E+07	9.31E-14	3.91E-28	0.00E+00	0.00E+00
f_9	6.83E-22	3.56E-42	2.44E-05	2.37E-10	7.06E-08	1.05E-14	1.07E+00	6.23E-01	1.88E-113	3.01E-225	1.60E-01	3.31E-02	3.70E-05	2.42E-09	0.00E+00	0.00E+00
f_{10}	1.26E-03	6.71E-08	2.85E+05	4.09E+10	2.03E+05	6.55E+10	8.45E+00	3.67E+02	5.09E-12	2.69E-24	1.74E+00	1.97E+01	2.03E-08	3.15E-18	0.00E+00	0.00E+00
f_{11}	2.10E+01	2.73E+01	4.35E+05	3.96E+09	1.89E+08	5.04E+15	1.77E+08	2.48E+17	3.12E-10	2.56E-20	1.33E+08	1.12E+16	5.57E+01	4.38E+03	0.00E+00	0.00E+00
f_{12}	0.00E+00	0.00E+00	4.29E+01	7.36E+01	7.19E+03	3.02E+06	3.15E+04	2.78E+08	0.00E+00	0.00E+00	1.42E+03	9.61E+06	5.04E+01	2.19E+02	0.00E+00	0.00E+00
f_{13}	6.62E+02	5.35E+02	7.21E+02	1.99E+03	1.01E+03	8.50E+03	1.01E+03	1.56E+05	7.19E-13	7.78E-25	1.24E+03	1.93E+04	4.40E+02	2.24E+03	0.00E+00	0.00E+00
f_{14}	5.18E+02	5.57E+02	5.11E+02	8.80E+02	8.36E+02	1.09E+04	1.60E+03	2.39E+05	3.85E-06	1.34E-10	1.37E+03	2.97E+04	3.14E+02	6.74E+02	0.00E+00	0.00E+00
f_{15}	1.29E-02	5.19E-06	2.98E+00	2.70E-02	1.63E+01	3.25E-01	2.01E+01	8.01E-03	1.99E+01	7.88E-04	1.99E+01	3.65E-02	1.95E+00	6.81E-02	8.88E-16	0.00E+00
f_{16}	6.56E-01	2.46E-01	3.79E+01	1.48E+01	6.46E+01	5.90E+01	5.66E+01	4.25E-02	2.30E-04	3.61E+01	1.06E+01	1.06E+02	5.76E-02	3.34E-02	0.00E+00	0.00E+00
f_{17}	5.23E-03	2.87E-06	1.82E-01	7.96E-04	8.57E+01	1.55E+03	4.21E+01	4.63E+03	9.99E-17	1.13E-33	9.00E+00	6.13E+02	1.77E-03	1.54E-05	0.00E+00	0.00E+00
f_{18}	2.76E+00	2.90E-02	4.71E+00	4.78E-02	1.68E+01	6.86E-01	7.37E+00	2.43E+02	4.10E-01	2.96E-03	2.61E+01	1.02E+02	2.03E+00	1.64E-01	0.00E+00	0.00E+00
f_{19}	9.36E-01	1.28E-02	1.91E+02	5.63E+01	2.11E+02	3.53E+01	5.20E+00	8.93E+01	4.36E-10	5.24E-20	8.66E+01	8.02E+01	3.83E+01	2.25E+01	0.00E+00	0.00E+00
f_{20}	2.33E+00	8.10E-01	2.41E+00	1.93E-01	7.60E+08	1.04E+17	1.22E+06	5.81E+12	7.63E-06	2.22E-12	6.39E+09	2.53E+20	7.42E+00	8.03E+00	6.75E-01	3.37E-03
f_{21}	2.06E+01	6.59E+01	1.86E+01	2.04E+01	1.66E+09	3.11E+17	1.83E+09	2.85E+19	1.07E-03	4.85E-08	7.49E+08	1.12E+19	1.02E+02	8.11E+03	1.91E+01	8.13E-02
f_{22}	1.20E+02	1.19E+02	6.11E+02	3.06E+03	1.34E+03	1.50E+03	4.44E+02	3.05E+05	1.87E-03	1.87E-07	1.59E+03	4.82E+03	7.95E+02	3.21E+03	0.00E+00	0.00E+00
f_{23}	5.67E-01	3.47E-02	1.27E+02	2.68E+01	2.19E+04	8.46E+06	7.35E+03	1.41E+08	1.03E-15	6.67E-31	5.08E+03	1.49E+08	3.32E+00	4.45E+00	0.00E+00	0.00E+00
f_{24}	1.69E+03	3.16E+05	1.38E+04	6.07E+06	1.46E+08	9.22E+14	4.75E+07	1.74E+16	1.95E-03	6.56E-03	3.73E+08	3.73E+18	3.65E+02	2.17E+04	1.98E+02	4.59E-02
w/!	0/1/2/3	1/0/2/3	0/0/2/4	0/0/2/4	0/0/2/4	0/0/2/4	0/0/2/4	0/0/2/4	3/1/2/0	0/0/2/4	0/0/2/4	0/0/2/4	0/0/2/4	0/0/2/4	0/0/2/4	0/0/2/4

TABLE 6. Rank-sum test results of SCBBO and other EAs with $D = 500$.

F	GWO		WOA		SSA		AEFA		AOA		ChOA		MPA		SCBBO	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std
f_1	8.21E-53	8.60E-53	9.51E-57	6.51E-112	3.62E-05	2.90E-09	4.47E+04	4.61E+03	3.37E+01	2.18E+01	5.47E-13	5.90E-13	1.98E-125	3.28E-125	0.00E+00	0.00E+00
f_2	9.77E-32	3.87E-32	7.99E-41	6.06E-80	6.64E-03	4.66E-05	2.25E+02	1.45E+02	7.77E-01	2.63E-01	1.16E-09	5.20E-10	1.09E-48	5.38E-48	0.00E+00	0.00E+00
f_3	2.68E-53	2.35E-53	3.19E-57	8.11E-113	6.70E-06	8.92E-11	1.70E+04	2.17E+03	1.76E+01	9.67E+00	2.75E-13	4.83E-13	1.91E-125	5.52E-125	0.00E+00	0.00E+00
f_4	3.58E+01	7.72E+00	2.03E-11	1.24E-21	1.24E-04	8.74E-09	1.95E+01	1.07E+00	2.20E+01	1.68E+00	9.76E+01	1.22E+00	5.50E-42	6.54E-42	0.00E+00	0.00E+00
f_5	1.09E-74	1.98E-74	5.27E-85	4.45E-168	7.70E-04	8.12E-07	6.25E+15	1.07E+15	8.56E+14	5.35E+14	1.12E-14	2.76E-14	1.22E-221	4.00E+00	0.00E+00	0.00E+00
f_6	5.46E-49	4.66E-49	8.84E-53	6.46E-104	4.95E-01	5.07E-01	5.98E+08	7.95E+07	2.56E+05	1.74E+05	4.64E-09	4.90E-09	3.06E-09	2.96E-121	0.00E+00	0.00E+00
f_7	1.12E-03	4.24E-04	2.20E-05	4.96E-10	1.63E-04	2.49E-08	4.46E+02	8.27E+01	1.77E+00	4.10E-01	1.92E-03	7.49E-04	2.44E-04	1.50E-04	1.45E-05	1.94E-10
f_8	3.04E-53	2.72E-53	8.64E-57	1.17E-111	2.14E-05	4.95E-09	1.65E+04	1.23E+03	1.33E+01	9.62E+00	3.58E-13	3.88E-13	1.07E-125	1.85E-125	0.00E+00	0.00E+00
f_9	3.99E-12	2.82E-11	0.00E+00	0.00E+00	1.94E-12	1.45E-23	2.96E-05	7.39E-05	9.00E-16	2.08E-15	1.80E+00	3.20E-01	0.00E+00	0.00E+00	0.00E+00	0.00E+00
f_{10}	3.18E-15	2.56E-16	0.00E+00	0.00E+00	4.36E-04	2.38E-07	2.07E+10	2.61E+10	5.39E-02	1.54E-02	1.20E-10	7.07E-11	0.00E+00	0.00E+00	0.00E+00	0.00E+00
f_{11}	4.38E-50	5.80E-50	8.18E-54	4.50E-106	6.92E-01	1.28E+00	5.45E+08	1.22E+08	3.18E+04	2.00E+04	2.38E-10	3.01E-10	5.01E-122	1.12E-121	0.00E+00	0.00E+00
f_{12}	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.84E+04	1.31E+03	3.63E+02	1.02E+02	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
f_{13}	0.00E+00	0.00E+00	0.00E+00	0.00E+00	7.27E-06	1.70E-10	1.83E+03	1.17E+02	1.35E+03	2.07E+02	1.94E-10	5.41E-10	0.00E+00	0.00E+00	0.00E+00	0.00E+00
f_{14}	0.00E+00	0.00E+00	0.00E+00	0.00E+00	9.30E-06	5.07E-10	1.83E+03	1.51E+02	2.18E+03	3.04E+02	7.00E-02	1.69E-01	0.00E+00	0.00E+00	0.00E+00	0.00E+00
f_{15}	5.38E-14	5.99E-15	1.49E+01	7.18E+01	1.00E-04	2.02E-08	6.92E+00	2.06E-01	2.24E+00	3.79E+00	2.00E+01	8.43E-03	4.44E-15	0.00E+00	8.88E-16	0.00E+00
f_{16}	2.60E-24	1.84E-23	5.89E-40	1.28E-78	2.33E-04	8.12E-08	1.08E+02	7.78E+00	9.40E+01	1.90E+01	1.69E-05	5.08E-05	1.45E-75	3.40E-75	0.00E+00	0.00E+00
f_{17}	1.29E-16	4.11E-17	0.00E+00	0.00E+00	3.02E-06	2.51E-11	2.91E+02	2.05E+01	8.50E-01	2.52E-01	3.74E-14	4.72E-14	0.00E+00	0.00E+00	0.00E+00	0.00E+00
f_{18}	3.04E-01	4.50E-02	2.09E-16	1.14E-32	7.34E-04	1.21E-06	1.83E+01	4.72E-01	1.06E+01	8.33E-01	1.33E-01	4.52E-02	1.64E-01	4.85E-02	0.00E+00	0.00E+00
f_{19}	6.37E-14	6.95E-14	0.00E+00	0.00E+00	2.81E-01	3.25E-02	2.82E+02	1.03E+01	2.28E+01	6.32E+00	4.87E-08	5.58E-08	0.00E+00	0.00E+00	0.00E+00	0.00E+00
f_{20}	7.07E-01	2.37E-02	1.07E+00	2.41E-04	7.75E-08	2.46E-14	1.90E+11	3.77E+10	4.55E+07	5.43E+07	1.08E+00	1.63E-02	3.18E-02	3.73E-03	8.17E-01	1.32E-03
f_{21}	5.94E+01	2.00E+01	8.47E+01	4.89E+02	1.75E-05	7.65E-10	2.47E+11	4.08E+10	2.24E+08	4.08E+08	9.79E+01	4.17E-01	3.83E+01	1.00E+00	4.93E+01	5.70E-02
f_{22}	2.46E-14	5.43E-15	1.16E-21	4.01E-42	7.96E+00	8.88E+00	1.73E+03	3.10E+02	2.14E+02	3.95E-03	3.42E-03	6.16E-44	9.96E-44	0.00E+00	0.00E+00	0.00E+00
f_{23}	0.00E+00	0.00E+00	0.00E+00	0.00E+00	2.42E-04	8.49E-08	4.89E+08	4.46E+03	1.70E+02	5.26E+01	5.40E-12	4.77E-12	0.00E+00	0.00E+00	0.00E+00	0.00E+00
f_{24}	4.98E+02	2.36E-01	4.99E+02	8.40E-02	4.12E-04	2.50E-07	2.25E+08	4.04E+07	4.74E+05	3.20E+05	4.99E+02	2.61E-01	4.91E+02	1.48E+02	4.98E+02	1.28E-02
w/t/	1/5/18	0/8/16	0/8/16	3/1/20	0/0/24	0/0/24	0/0/24	0/0/24	0/0/24	0/0/24	0/1/23	0/1/23	3/8/13	3/8/13		

TABLE 7. Results obtained by SCBBO for 24 benchmark functions with 500, 1000, 2000, 5000, 10000 dimensions.

F	SCBBO(D=500)		SCBBO(D=1000)		SCBBO(D=2000)		SCBBO(D=5000)		SCBBO(D=10000)	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std
f1	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
f2	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
f3	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
f4	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
f5	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
f6	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
f7	1.45E-05	1.94E-10	1.62E-05	3.53E-10	1.77E-05	4.87E-10	3.12E-05	4.94E-10	3.79E-05	2.06E-09
f8	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
f9	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
f10	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
f11	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
f12	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
f13	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
f14	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
f15	8.88E-16	0.00E+00	8.88E-16	0.00E+00	8.88E-16	0.00E+00	8.88E-16	0.00E+00	8.88E-16	0.00E+00
f16	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
f17	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
f18	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
f19	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
f20	8.17E-01	1.32E-03	9.54E-01	4.62E-03	1.13E+00	1.05E-04	2.36E+00	9.74E-03	1.63E+00	7.58E-02
f21	4.93E+01	1.70E-02	4.96E+01	1.70E-02	9.96E+01	1.96E-02	2.00E+02	9.98E-02	5.00E+02	4.45E-01
f22	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
f23	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
f24	4.98E+02	1.28E-02	9.98E+02	7.43E-03	2.00E+03	4.59E-03	5.00E+03	6.42E-03	1.00E+04	7.76E-02

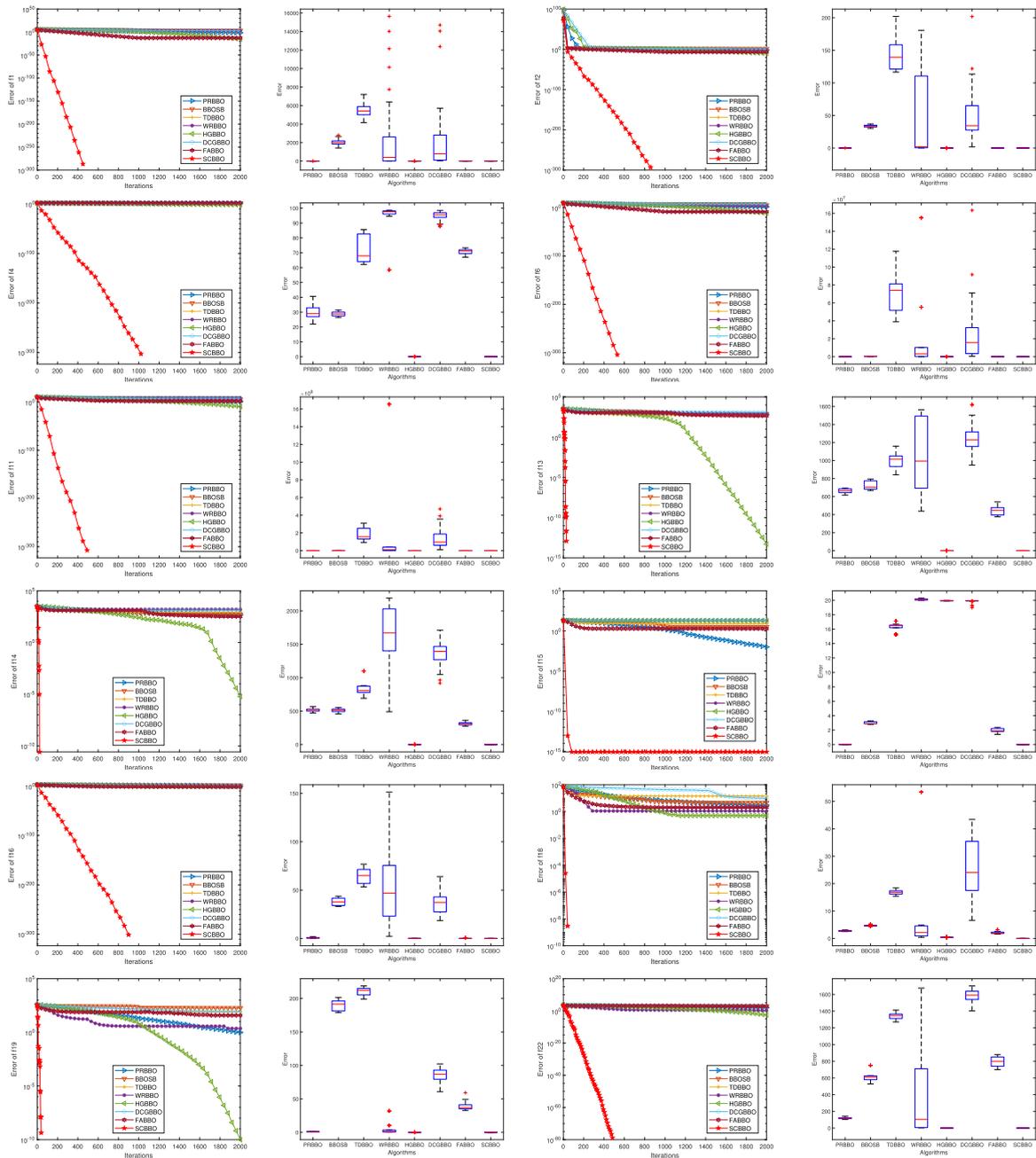


FIGURE 11. Convergence curves and boxplots of scbbo and seven bbo variants on different benchmark functions with $d = 200$.

for algorithms. An algorithm should converge quickly and be able to solve high dimensional problems. To further analyze and compare the performance of the proposed SCBBO on high-dimensional global optimization problems, SCBBO is optimized in the high-dimensional environment of $D = 1000$, $D = 2000$, $D = 5000$ and $D = 10000$, respectively. Similarly, to avoid contingency, the algorithm runs 50 times on each benchmark function, and the mean and standard deviation of the 50 errors are recorded. As shown in TABLE 7, to facilitate comparison, the results obtained by SCBBO on 500 dimensions are also included.

From TABLE 7, SCBBO can still converge precisely on 10000 dimensions, and the error obtained on 19 benchmark functions are 0. Except for the multi-modal function f24, the solution accuracy of SCBBO is basically unchanged on different high dimensions. Furthermore, FIGURE 13 shows the convergence curves of SCBBO on some benchmark functions with different dimensions. It can be seen that with the increase of dimensions, the convergence curves of SCBBO are basically the same. Therefore, the algorithm proposed in this paper has powerful searching ability, and its performance is basically not affected by dimensions,

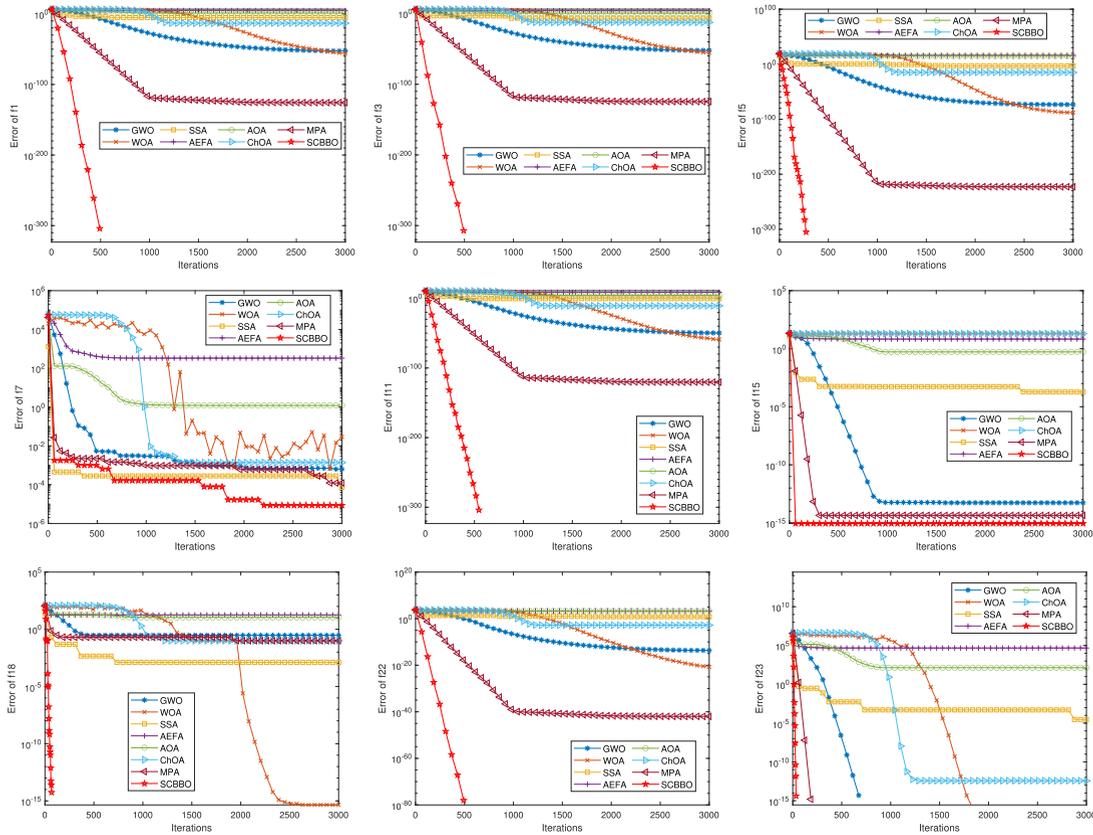


FIGURE 12. Convergence curves of SCBBO and other seven EAs on different benchmark functions with $D = 500$.

which can effectively solve the high-dimensional global optimization problems.

VII. APPLICATION OF SCBBO ON ENGINEERING DESIGN PROBLEMS

This section further verifies the effectiveness and advancement of SCBBO by solving three constrained real optimization problems in engineering design (pressure vessel design, tension/compression spring design and welded beam design). These engineering design problems have been extensively studied and solved in many literatures, so we selected the methods and research results in the past few years for comparison to better clarify the performance of SCBBO. In SCBBO, the population size NP is 50, and the maximum number of iterations $MaxIt$ only needs 100.

A. PRESSURE VESSEL DESIGN

The goal of the pressure vessel design problem is to minimize the cost of fabrication (pairing, molding and welding). The design of the pressure vessel is shown in FIGURE 14. Both ends of the pressure vessel are capped with a cap, and the cap at one end of the head is hemispherical. L is the section length of the cylinder part without considering the head, R is the inner wall radius of the cylinder part, T_s and T_h are the wall thickness of the cylinder part and the

head respectively. Therefore, L, R, T_s and T_h are the four optimization variables for the pressure vessel design problem. The objective function and four optimization constraints of the problem are expressed as follows:

$$\begin{cases}
 X = [x_1, x_2, x_3, x_4] = [T_s, T_h, R, L] \\
 \text{Minimize } f(X) = 0.6224 x_1 x_3 x_4 + \\
 1.7781 x_2 x_3^2 + 3.1661 x_1^2 x_4 + 19.84 x_1^2 x_3 \\
 \text{S.t. } g_1(X) = -x_1 + 0.0193 x_3 \leq 0 \\
 g_2(X) = -x_2 + 0.00954 x_3 \leq 0 \\
 g_3(X) = -\pi x_3^2 x_4 - 4\pi x_3^3 / 3 + 1296000 \leq 0 \\
 g_4(X) = x_4 - 240 \leq 0 \\
 \text{Where } 0 \leq x_i \leq 100, i = 1, 2; \\
 10 \leq x_i \leq 200, i = 3, 4
 \end{cases} \quad (23)$$

We apply SCBBO to solve pressure vessel design problem, and compare its results with 16 excellent algorithms proposed in the past decade, as shown in TABLE 8. As can be seen from TABLE 8, the results obtained by SCBBO in pressure vessel design are superior to other comparison algorithms, and the cost is minimal.

B. TENSION/COMPRESSION SPRING DESIGN

Spring is an important part in industrial production, and there are many factors that affect the structural performance

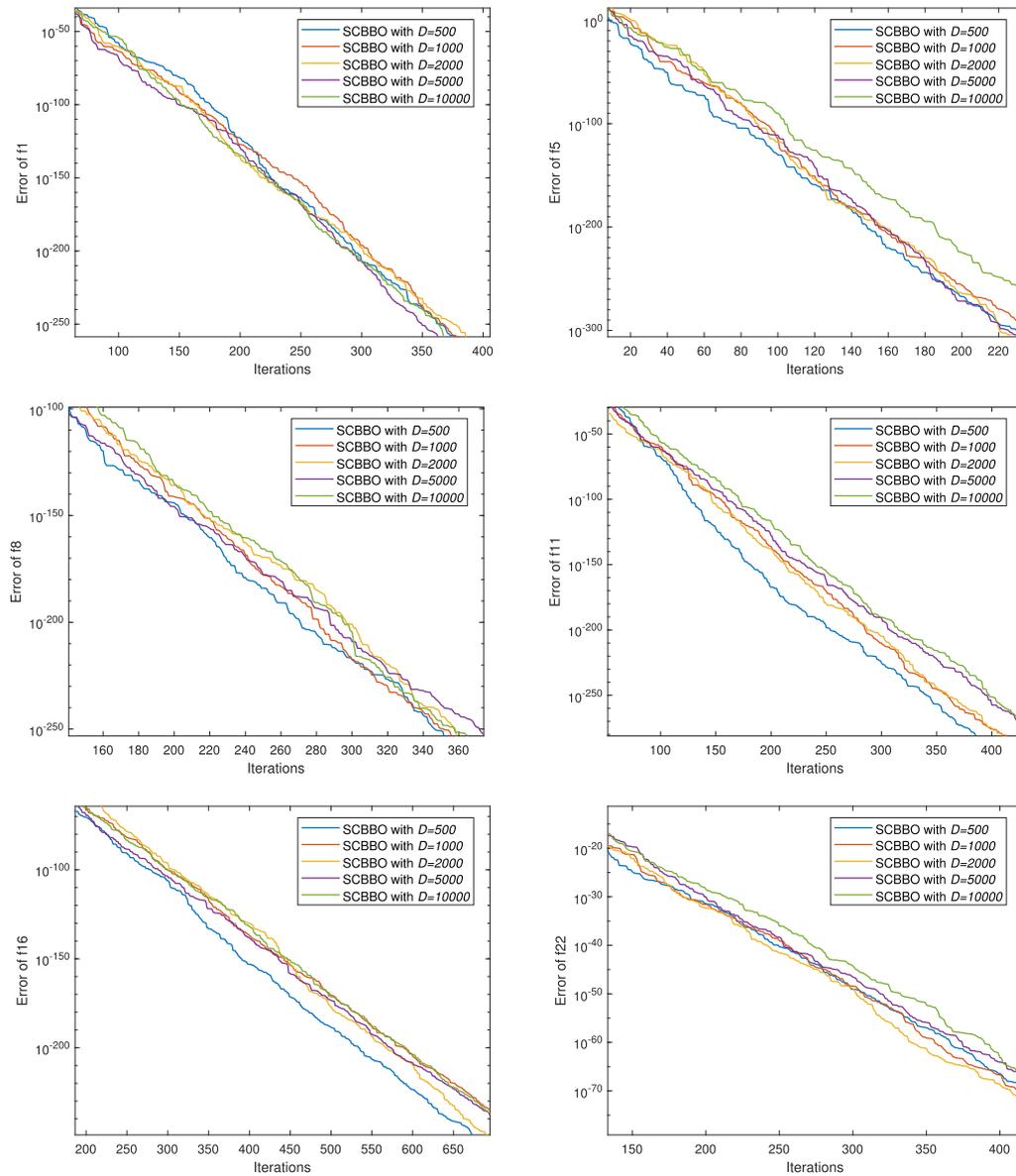


FIGURE 13. Convergence curves of SCBBO with $D = 500, 1000, 2000, 5000$ and 10000 on some benchmark functions.

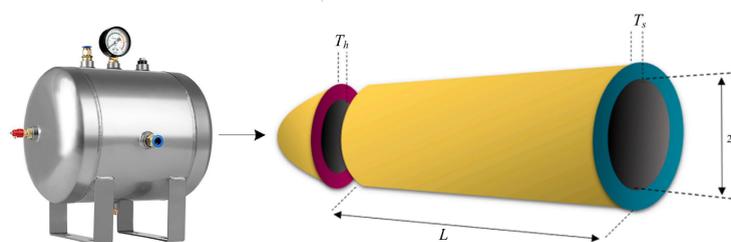


FIGURE 14. Schematic view of pressure vessel design problem.

of it. As shown in FIGURE 15, the design problem of the tension/compression spring is to minimize the weight of the spring while meeting the constraints of minimum deflection, vibration frequency, and shear stress. The problem consists

of three continuous decision variables: the wire diameter (d), the mean coil diameter (D) and the number of active coils (P). Through the relationship between design parameters and performance of spring, the following mathematical model can

TABLE 8. The experimental results of SCBBO and other algorithms for pressure vessel design problem.

Time	Algorithm	Optimal values for variables				Optimal cost
		T_s	T_h	R	L	
2012	CGDA	0.8125	0.4375	42.0975	176.6484	6059.8391
2013	BA	0.8125	0.4375	42.0984	179.6366	6059.7143
2013	UABC	0.8125	0.4375	42.098446	176.636596	6059.714335
2014	GWO	0.8125	0.4345	42.089181	176.758731	6051.5639
2015	AFA	0.8125	0.4375	42.0984	176.6366	6059.7143
2015	MFO	0.8125	0.4375	42.098445	176.636596	6059.7143
2016	SCA	0.817577	0.417932	41.74939	183.5727	6137.3724
2016	WOA	0.8125	0.4375	42.0982699	176.638998	6059.7410
2017	TEO	0.8125	0.4325	42.0984	173.6366	6059.71
2018	EEGWO	13.09291	6.792196	42.09758	176.6495	6059.8704
2019	ISCA [48]	12.96419	7.150134	42.09829	176.6392	6059.7489
2019	L-SHADE	1.7913	0.6373	61.4220	29.4625	10502.06
2019	HGSO [71]	1.272354	0.624693	65.46899	10	7433.4480
2021	I-GWO [72]	0.779031	0.385501	40.36313	199.4017	5888.3400
2021	RSO [73]	0.775967	0.383127	40.313297	200.00000	5878.5395
2021	AVOA [74]	0.778954	0.3850374	40.360312	199.434299	5886.676593
(this work)	SCBBO	0.7885488	0.3910279	40.84456	192.8193	5859.3088

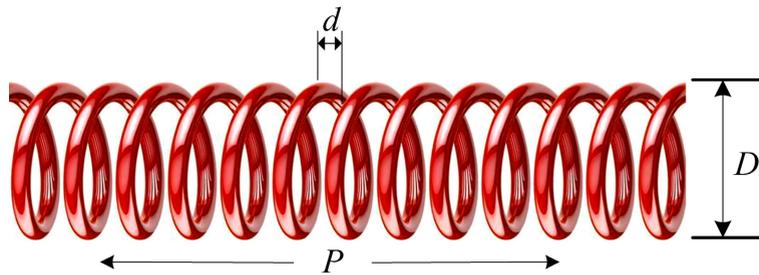


FIGURE 15. Schematic view of tension/compression spring design problem.

TABLE 9. The experimental results of SCBBO and other algorithms for tension/compression spring design problem.

Time	Algorithm	Optimal values for variables			Optimal cost
		d	D	P	
2012	CGDA	0.0516925	0.3568108	11.2835059	0.012665
2013	BA	0.05169	0.35673	11.2885	0.0126652
2013	UABC	0.051691	0.356769	11.285988	0.012665
2014	GWO	0.05169	0.356737	11.28885	0.012666
2015	AFA	0.051667	0.356198	11.319561	0.0126653
2015	MFO	0.051994457	0.36410932	10.868421862	0.0126669
2016	SCA	0.05078	0.334779	12.72269	0.0127097
2016	WOA	0.051207	0.345215	12.54854	0.0126763
2017	TEO	0.051775	0.358792	11.168390	0.012665
2018	EEGWO	0.051673	0.35634	11.3113	0.012665
2019	ISCA [48]	0.0520217	0.364768	10.8323	0.012667
2019	L-SHADE	0.0839	0.93420	4.50460	0.04275
2019	HHO	0.0570	0.4991	6.2180	0.01281
2021	RSO [73]	0.051075	0.341987	12.0667	0.012655697
2021	AVOA [74]	0.051669833	0.356255347	11.316126	0.012665240
2022	HBA [75]	0.0506	0.3552	11.373	0.01207
(this work)	SCBBO	0.0522106	0.369395	10.763	0.009765

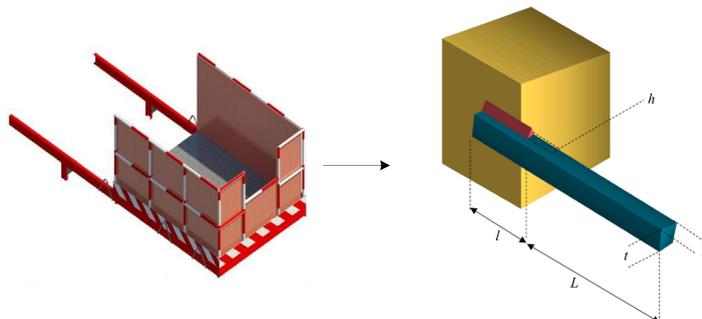


FIGURE 16. Schematic view of welded beam design problem.

TABLE 10. The experimental results of SCBBO and other algorithms for welded beam design problem.

Time	Algorithm	Optimal values for variables				Optimal cost
		<i>h</i>	<i>l</i>	<i>t</i>	<i>b</i>	
2008	DEDS	0.2444	6.2175	8.2915	0.2444	2.3810
2009	EMEA	0.2443	6.2201	8.2940	0.2444	2.3816
2012	KH	0.150197	5.31408	9.045851	0.205745	1.861801
2014	ISA	0.2443	6.2199	8.2915	0.2443	2.3812
2016	GWO	0.205409	3.478839	9.035941	0.205774	1.725700
2016	WOA	0.189953	3.99627	8.71047	0.227041	1.871528
2016	EHO	0.4834	2.4950	4.4538	0.8488	2.3234
2018	MTSA	0.24415742	6.22306595	8.29555011	0.24440474	2.38241101
2019	ISCA [48]	0.24415742	6.2178	8.2919	0.24437	2.3810
2019	HGSO [71]	0.2054	3.4476	9.0269	0.2060	1.7260
2019	L-SHADE	0.4819	3.2140	5.4763	0.5753	3.43372
2019	HHO	0.1961	3.7449	9.0061	0.2071	1.75163
2021	I-GWO [72]	0.20573	3.47049	9.036624	0.20573	1.724853
2021	RSO [73]	0.205397	3.465789	9.034571	0.201097	1.722789
2021	AVOA [74]	0.205730	3.470474	9.036621	0.205730	1.724852
2022	HBA [75]	0.2057	3.4704	9.0366	0.2057	1.72451
(this work)	SCBBO	0.205398	3.258989	9.038399	0.205801	1.696157

be obtained:

$$\begin{cases}
 X = [x_1, x_2, x_3] = [d, D, P] \\
 \text{Minimize } f(X) = (x_3 + 2)x_2x_1^2 \\
 \text{St. } g_1(X) = 1 - \frac{x_2^3x_3}{71785x_1^4} \leq 0 \\
 g_2(X) = \frac{4x_2^2 - x_1x_2}{12566(x_2x_1^3 - x_1^4)} + \frac{1}{5108x_1^2 - 1} \leq 0 \\
 g_3(X) = 1 - \frac{140.45x_1}{x_2^2x_3} \leq 0 \\
 g_4(X) = \frac{x_1 + x_2}{1.5} - 1 \leq 0 \\
 \text{Where } 0.05 \leq x_1 \leq 2, 0.25 \leq x_2 \leq 1.3, 2 \leq x_3 \leq 15.
 \end{cases} \tag{24}$$

The objective function in Eq. (24) is a typical multi-variable constrained optimization problem. In order to obtain better spring design parameters, many scholars applied the improved EAs to this engineering optimization problem. To illustrate the design effect of SCBBO, 16 outstanding algorithms of nearly ten years are also applied to the objective function and the optimization results are compared. The spring design parameters and objective function results obtained by each algorithm are shown in TABLE 9. As can be seen, the results obtained by SCBBO in the design of tension/compression spring are better than other EAs, and the spring weight is the minimum.

C. WELDING BEAM DESIGN

The goal of the welding beam design problem is to obtain the minimum manufacturing cost. As shown in FIGURE 16, four design variables of welded beam need to be optimized: the thickness of weld (*h*), length of attached part of bar (*l*), height of bar (*t*), and the thickness of bar (*b*). While optimizing the design variables, ensure that the seven constraints are not violated. These constraints are shear stress (τ), bending stress

in beam (σ), deflection of beam end (δ) and buckling load of bar (P_b). The mathematical model of welding beam design is as follows:

$$\begin{cases}
 X = [x_1, x_2, x_3, x_4] = [h, l, t, b] \\
 \text{Minimize } f(X) = 1.10471 x_1^2x_2 \\
 + 0.04811 x_3x_4 \cdot (14.0 + x_2) \\
 \text{S.t. } g_1(X) = \tau(X) - \tau_{\max} \leq 0 \\
 g_2(X) = \sigma(X) - \sigma_{\max} \leq 0 \\
 g_3(X) = \delta(X) - \delta_{\max} \leq 0 \\
 g_4(X) = x_1 - x_4 \leq 0 \\
 g_5(X) = P - P_c(X) \leq 0 \\
 g_6(X) = 0.125 - x_1 \leq 0 \\
 g_7(X) = 1.10471 x_1^2 \\
 + 0.04811 x_3x_4 \cdot (14.0 + x_2) - 0.5 \leq 0 \\
 \text{Where } 0.1 \leq x_1, x_2 \leq 2, 0.1 \leq x_3, x_4 \leq 10. \\
 \tau(\vec{x}) = \sqrt{(\tau')^2 + (\tau'')^2 + (l\tau'\tau'') / \sqrt{0.25(l^2 + (h+t)^2)}} \\
 \tau' = \frac{6,000}{\sqrt{2}hl}, \quad \sigma(\vec{x}) = \frac{504,000}{t^2b}, \quad \delta(\vec{x}) = \frac{65,856,000}{(30 \times 10^6)bt^3}, \\
 \tau'' = \frac{6,000(14 + 0.5l)\sqrt{0.25(l^2 + (h+t)^2)}}{2[0.707hl(l^2/12 + 0.25(h+t)^2)]} \\
 P_c(\vec{x}) = 64,746.022(1 - 0.0282346t)tb^3
 \end{cases} \tag{25}$$

SCBBO is applied to solve the welding beam design problem and compared with 16 optimization algorithms. as shown in TABLE 10, SCBBO has the best optimization result, and the results of GWO, HGSO, I-GWO, RSO, AVOA and HBA are similar to SCBBO.

In conclusion, SCBBO achieves better optimization results than other EAs on three complex engineering design optimization problems. Therefore, SCBBO can be widely

used in constrained optimization problems and is an advanced algorithm worthy of being popularized and adopted.

VIII. CONCLUSION

In order to improve the performance of BBO for high-dimensional global optimization functions, this paper proposes a new BBO variant based on sine cosine algorithm and dual learning strategy, named SCBBO. The uniqueness and innovation of this paper can be summarized as follows: (a). This paper uses Latin hypercube sampling method to generate initial population, which improves ergodicity of population distribution. At the same time, the shortcomings of the position updating formula of SCA are analyzed. Then the nonlinear transformation parameters and inertia weight adjusting factor are designed, which are combined with the original BBO's migration operator to obtain a hybrid migration operator that adjusts the search state with the iterations. (b). By combining Laplacian random number and Gaussian random number, a dynamic hybrid mutation operator is obtained, and the dual learning strategy is integrated into BBO, which effectively balances the exploration and exploitation of the algorithm and helps it improve the convergence speed and accuracy. (c). A sequence convergence model is established to prove that SCBBO has global convergence, and the computational complexity of SCBBO is analyzed by comparing with the original BBO. (d). 24 benchmark functions are used for comparative simulation experiments, and the results prove that the ability of SCBBO to solve global optimization problems can reach 10000 dimensions. In engineering design optimization problems, SCBBO can obtain better design parameters, which shows that SCBBO has higher practical application value.

In the future work, SCBBO can be combined with more complex optimization problems in other fields, such as image processing, neural networks, support vector machines, etc.

AUTHORSHIP CONTRIBUTION STATEMENT

Ziyu Zhang: investigation, methodology, experiment, and writing: original draft. Yuelin Gao: supervision and funding acquisition. Wenlu Zuo: project administration and writing: review and editing.

STATEMENTS AND DECLARATIONS

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this article. All authors guarantee that the article is legitimate and belongs to their own scientific research results. No copying, plagiarism, infringement, data forgery, and other bad behavior, not involve state secrets. The source of others' work quoted in the manuscript has been indicated. The authors will be responsible for any economic losses and negative social impacts caused by bad behavior themselves.

REFERENCES

- [1] P. K. R. Maddikunta, T. R. Gadekallu, R. Kaluri, G. Srivastava, R. M. Parizi, and M. S. Khan, "Green communication in IoT networks using a hybrid optimization algorithm," *Comput. Commun.*, vol. 159, pp. 97–107, Jun. 2020, doi: [10.1016/j.comcom.2020.05.020](https://doi.org/10.1016/j.comcom.2020.05.020).
- [2] B. Chen, Y. Lin, W. Zeng, H. Xu, and D. Zhang, "The mean-variance cardinality constrained portfolio optimization problem using a local search-based multi-objective evolutionary algorithm," *Int. J. Speech Technol.*, vol. 47, no. 2, pp. 505–525, Sep. 2017, doi: [10.1007/s10489-017-0898-z](https://doi.org/10.1007/s10489-017-0898-z).
- [3] X. Zhang, Y. Chen, Y. Wang, R. Ding, Y. Zheng, Y. Wang, X. Zha, and X. Cheng, "Reactive voltage partitioning method for the power grid with comprehensive consideration of wind power fluctuation and uncertainty," *IEEE Access*, vol. 8, pp. 124514–124525, 2020, doi: [10.1109/ACCESS.2020.3004484](https://doi.org/10.1109/ACCESS.2020.3004484).
- [4] C. Liu, Y. Yu, X. Li, and P. Wang, "Application of entity relation extraction method under CRF and syntax analysis tree in the construction of military equipment knowledge graph," *IEEE Access*, vol. 8, pp. 200581–200588, 2020, doi: [10.1109/ACCESS.2020.3034894](https://doi.org/10.1109/ACCESS.2020.3034894).
- [5] K. Lu, G. Zeng, and W. Zhou, "Adaptive constrained population extremal optimisation-based robust proportional-integral-derivation frequency control method for an islanded microgrid," *IET Cyber-Syst. Robot.*, vol. 3, no. 3, pp. 210–227, Sep. 2021, doi: [10.1049/csy2.12028](https://doi.org/10.1049/csy2.12028).
- [6] G. Q. Zeng, X.-Q. Xie, M.-R. Chen, and J. Weng, "Adaptive population extremal optimization-based PID neural network for multivariable nonlinear control systems," *Swarm Evol. Comput.*, vol. 44, pp. 320–334, Feb. 2019, doi: [10.1016/j.swevo.2018.04.008](https://doi.org/10.1016/j.swevo.2018.04.008).
- [7] D. Simon, "Biogeography-based optimization," *IEEE Trans. Evol. Comput.*, vol. 12, no. 6, pp. 702–713, Dec. 2008, doi: [10.1109/TEVC.2008.919004](https://doi.org/10.1109/TEVC.2008.919004).
- [8] H. Ma, "An analysis of the equilibrium of migration models for biogeography-based optimization," *Inf. Sci.*, vol. 180, no. 18, pp. 3444–3464, Sep. 2010, doi: [10.1016/j.ins.2010.05.035](https://doi.org/10.1016/j.ins.2010.05.035).
- [9] L. Wei, N. Wang, and H. Lu, "A novel BBO algorithm based on local search and nonuniform variation for iris classification," *Complexity*, vol. 2021, pp. 1–17, Apr. 2021, doi: [10.1155/2021/6694695](https://doi.org/10.1155/2021/6694695).
- [10] Q. X. Feng, S. Liu, J. Zhang, G. Yang, and L. Yong, "Improved biogeography-based optimization with random ring topology and Powell's method," *Appl. Math. Model.*, vol. 41, pp. 630–649, Jan. 2016, doi: [10.1016/j.apm.2016.09.020](https://doi.org/10.1016/j.apm.2016.09.020).
- [11] X. Zhang, Q. Kang, Q. Tu, J. Cheng, and X. Wang, "Efficient and merged biogeography-based optimization algorithm for global optimization problems," *Soft Comput.*, vol. 23, no. 12, pp. 4483–4502, Jun. 2019, doi: [10.1007/s00500-018-3113-1](https://doi.org/10.1007/s00500-018-3113-1).
- [12] Y. An, X. Chen, Y. Li, Y. Han, J. Zhang, and H. Shi, "An improved non-dominated sorting biogeography-based optimization algorithm for the (hybrid) multi-objective flexible job-shop scheduling problem," *Appl. Soft Comput.*, vol. 99, Feb. 2021, Art. no. 106869, doi: [10.1016/j.asoc.2020.106869](https://doi.org/10.1016/j.asoc.2020.106869).
- [13] G. Xiong, Y. Li, J. Chen, D. Shi, and X. Duan, "Polyphyletic migration operator and orthogonal learning aided biogeography-based optimization for dynamic economic dispatch with valve-point effects," *Energy Convers. Manage.*, vol. 80, pp. 457–468, Apr. 2014, doi: [10.1016/j.enconman.2013.12.052](https://doi.org/10.1016/j.enconman.2013.12.052).
- [14] J. C. Bansal and P. Farswan, "A novel disruption in biogeography-based optimization with application to optimal power flow problem," *Int. J. Speech Technol.*, vol. 46, no. 3, pp. 590–615, Apr. 2017, doi: [10.1007/s10489-016-0848-1](https://doi.org/10.1007/s10489-016-0848-1).
- [15] X. Li and M. Yin, "Multi-operator based biogeography based optimization with mutation for global numerical optimization," *Comput. Math. With Appl.*, vol. 64, no. 9, pp. 2833–2844, Nov. 2012, doi: [10.1016/j.camwa.2012.04.015](https://doi.org/10.1016/j.camwa.2012.04.015).
- [16] M. Ergezer, "Oppositional biogeography-based optimization," Cleveland State Univ., Cleveland, OH, USA, Tech. Rep., 2014, doi: [10.1109/ICSMC.2009.5346043](https://doi.org/10.1109/ICSMC.2009.5346043).
- [17] A. Reihanian, M.-R. Feizi-Derakhshi, and H. S. Aghdasi, "NBBO: A new variant of biogeography-based optimization with a novel framework and a two-phase migration operator," *Inf. Sci.*, vol. 504, pp. 178–201, Dec. 2019, doi: [10.1016/j.ins.2019.07.054](https://doi.org/10.1016/j.ins.2019.07.054).
- [18] G. Wang, L. Guo, H. Duan, H. Wang, L. Liu, and M. Shao, "Hybridizing harmony search with biogeography based optimization for global numerical optimization," *J. Comput. Theor. Nanosci.*, vol. 10, no. 10, pp. 2312–2322, 2013, doi: [10.1166/jctn.2013.3207](https://doi.org/10.1166/jctn.2013.3207).

- [19] K. S. Lee and Z. W. Geem, "A new meta-heuristic algorithm for continuous engineering optimization: Harmony search theory and practice," *Comput. Methods Appl. Mech. Eng.*, vol. 194, nos. 36–38, pp. 3902–3933, Sep. 2005, doi: [10.1016/j.cma.2004.09.007](https://doi.org/10.1016/j.cma.2004.09.007).
- [20] Y.-J. Zheng, H.-F. Ling, H.-H. Shi, H.-S. Chen, and S.-Y. Chen, "Emergency railway wagon scheduling by hybrid biogeography-based optimization," *Comput. Oper. Res.*, vol. 43, no. 1, pp. 1–8, 2014, doi: [10.1016/j.cor.2013.09.002](https://doi.org/10.1016/j.cor.2013.09.002).
- [21] G.-G. Wang, A. H. Gandomi, and A. H. Alavi, "An effective krill herd algorithm with migration operator in biogeography-based optimization," *Appl. Math. Model.*, vol. 38, nos. 9–10, pp. 2454–2462, May 2014, doi: [10.1016/j.apm.2013.10.052](https://doi.org/10.1016/j.apm.2013.10.052).
- [22] W. L. Lim, A. Wibowo, M. I. Desa, and H. Haron, "A biogeography-based optimization algorithm hybridized with tabu search for the quadratic assignment problem," *Comput. Intell. Neurosci.*, vol. 2016, pp. 1–12, Jan. 2016, doi: [10.1155/2016/5803893](https://doi.org/10.1155/2016/5803893).
- [23] A. Misevicius, "An implementation of the iterated Tabu search algorithm for the quadratic assignment problem," *OR Spectr.*, vol. 34, no. 3, pp. 665–690, Jul. 2012, doi: [10.1007/s00291-011-0274-z](https://doi.org/10.1007/s00291-011-0274-z).
- [24] C. K. Yogesh, M. Hariharan, R. Ngadiran, A. H. Adom, S. Yaacob, C. Berkai, and K. Polat, "A new hybrid PSO assisted biogeography-based optimization for emotion and stress recognition from speech signal," *Expert Syst. Appl.*, vol. 69, pp. 149–158, Mar. 2017, doi: [10.1016/j.eswa.2016.10.035](https://doi.org/10.1016/j.eswa.2016.10.035).
- [25] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. Int. Conf. Neural Netw. (ICNN)*, vol. 4, 1995, pp. 1942–1948, doi: [10.1109/ICNN.1995.488968](https://doi.org/10.1109/ICNN.1995.488968).
- [26] X. Zhang, Q. Kang, J. Cheng, and X. Wang, "A novel hybrid algorithm based on biogeography-based optimization and grey wolf optimizer," *Appl. Soft Comput.*, vol. 67, pp. 197–214, Jun. 2018, doi: [10.1016/j.asoc.2018.02.049](https://doi.org/10.1016/j.asoc.2018.02.049).
- [27] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Adv. Eng. Softw.*, vol. 69, pp. 46–61, Mar. 2014, doi: [10.1016/j.advengsoft.2013.12.007](https://doi.org/10.1016/j.advengsoft.2013.12.007).
- [28] P. Farswan and J. C. Bansal, "Fireworks-inspired biogeography-based optimization," *Soft Comput.*, vol. 23, no. 16, pp. 7091–7115, Aug. 2019, doi: [10.1007/s00500-018-3351-2](https://doi.org/10.1007/s00500-018-3351-2).
- [29] Y. Tan and Y. Zhu, "Fireworks algorithm for optimization," in *Proc. Int. Conf. Swarm Intell.* Berlin, Germany: Springer, 2010, pp. 355–364, doi: [10.1007/978-3-642-13495-1_44](https://doi.org/10.1007/978-3-642-13495-1_44).
- [30] H. F. Ghatte, "A hybrid of firefly and biogeography-based optimization algorithms for optimal design of steel frames," *Arabian J. Sci. Eng.*, vol. 46, no. 5, pp. 4703–4717, May 2021, doi: [10.1007/s13369-020-05118-w](https://doi.org/10.1007/s13369-020-05118-w).
- [31] X. S. Yang, "Firefly algorithms for multimodal optimization," in *Proc. Int. Symp. Stochastic Algorithms.* Berlin, Germany: Springer, 2009, pp. 169–178, doi: [10.1007/978-3-642-04944-6_14](https://doi.org/10.1007/978-3-642-04944-6_14).
- [32] W. Gong, Z. Cai, C. X. Ling, and H. Li, "A real-coded biogeography-based optimization with mutation," *Appl. Math. Comput.*, vol. 216, no. 9, pp. 2749–2758, Jul. 2010, doi: [10.1016/j.amc.2010.03.123](https://doi.org/10.1016/j.amc.2010.03.123).
- [33] L.-L. Li, Y.-F. Yang, C.-H. Wang, and K.-P. Lin, "Biogeography-based optimization based on population competition strategy for solving the substation location problem," *Expert Syst. Appl.*, vol. 97, pp. 290–302, May 2018, doi: [10.1016/j.eswa.2017.12.039](https://doi.org/10.1016/j.eswa.2017.12.039).
- [34] R. Pal and M. Saraswat, "Histopathological image classification using enhanced bag-of-feature with spiral biogeography-based optimization," *Int. J. Speech Technol.*, vol. 49, no. 9, pp. 3406–3424, Sep. 2019, doi: [10.1007/s10489-019-01460-1](https://doi.org/10.1007/s10489-019-01460-1).
- [35] O. Rostami and M. Kaveh, "Optimal feature selection for SAR image classification using biogeography-based optimization (BBO), artificial bee colony (ABC) and support vector machine (SVM): A combined approach of optimization and machine learning," *Comput. Geosci.*, vol. 25, no. 3, pp. 911–930, Jun. 2021, doi: [10.1007/s10596-020-10030-1](https://doi.org/10.1007/s10596-020-10030-1).
- [36] M. Harrabi, O. B. Driss, and K. Ghedira, "A hybrid evolutionary approach to job-shop scheduling with generic time lags," *J. Scheduling*, vol. 24, no. 3, pp. 329–346, Jun. 2021, doi: [10.1007/s10951-021-00683-w](https://doi.org/10.1007/s10951-021-00683-w).
- [37] S. A. Jalae, A. Shakibaei, H. R. Horry, H. Akbarifard, A. GhasemiNejad, F. N. Robati, and N. A. Zarin, "A new hybrid metaheuristic method based on biogeography-based optimization and particle swarm optimization algorithm to estimate money demand in Iran," *MethodsX*, vol. 8, Jan. 2021, Art. no. 101226, doi: [10.1016/j.mex.2021.101226](https://doi.org/10.1016/j.mex.2021.101226).
- [38] J. Taghizadeh, M. Ghobaei-Arani, and A. Shahidinejad, "An efficient data replica placement mechanism using biogeography-based optimization technique in the fog computing environment," *J. Ambient Intell. Humanized Comput.*, pp. 1–21, Sep. 2021, doi: [10.1007/s12652-021-03495-0](https://doi.org/10.1007/s12652-021-03495-0).
- [39] W. Guo, M. Chen, L. Wang, Y. Mao, and Q. Wu, "A survey of biogeography-based optimization," *Neural Comput. Appl.*, vol. 28, no. 8, pp. 1909–1926, Sep. 2017, doi: [10.1007/s00521-016-2179-x](https://doi.org/10.1007/s00521-016-2179-x).
- [40] H. Ma, D. Simon, P. Siarry, Z. Yang, and M. Fei, "Biogeography-based optimization: A 10-year review," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 1, no. 5, pp. 391–407, Oct. 2017, doi: [10.1109/TETCI.2017.2739124](https://doi.org/10.1109/TETCI.2017.2739124).
- [41] J. S. Wang and J. D. Song, "Chaotic biogeography-based optimization algorithm," *IAENG Int. J. Comput. Sci.*, vol. 44, no. 2, pp. 122–134, 2017.
- [42] A. Mistri, P. K. Roy, and B. Mandal, "Chaotic biogeography-based optimization (CBBO) algorithm applied to economic load dispatch problem," in *Proc. Nat. Conf. Emerg. Trends Sustain. Technol. Eng. Appl. (NCETSTE)*, Feb. 2020, pp. 1–5, doi: [10.1109/NCETSTE48365.2020.9119956](https://doi.org/10.1109/NCETSTE48365.2020.9119956).
- [43] K. Burrage, P. Burrage, D. Donovan, and B. Thompson, "Populations of models, experimental designs and coverage of parameter space by Latin hypercube and orthogonal sampling," *Proc. Comput. Sci.*, vol. 51, pp. 1762–1771, Jan. 2015, doi: [10.1016/j.procs.2015.05.383](https://doi.org/10.1016/j.procs.2015.05.383).
- [44] Z. Zhao, J. Yang, Z. Hu, and H. Che, "A differential evolution algorithm with self-adaptive strategy and control parameters based on symmetric Latin hypercube design for unconstrained optimization problems," *Eur. J. Oper. Res.*, vol. 250, no. 1, pp. 30–45, Apr. 2016, doi: [10.1016/j.ejor.2015.10.043](https://doi.org/10.1016/j.ejor.2015.10.043).
- [45] D. Donovan, K. Burrage, P. Burrage, T. A. McCourt, B. Thompson, and E. Ş. Yazıcı, "Estimates of the coverage of parameter space by Latin hypercube and orthogonal array-based sampling," *Appl. Math. Model.*, vol. 57, pp. 553–564, May 2018, doi: [10.1016/j.apm.2017.11.036](https://doi.org/10.1016/j.apm.2017.11.036).
- [46] F. Zhao, S. Qin, Y. Zhang, W. Ma, C. Zhang, and H. Song, "A two-stage differential biogeography-based optimization algorithm and its performance analysis," *Expert Syst. Appl.*, vol. 115, pp. 329–345, Jan. 2019, doi: [10.1016/j.eswa.2018.08.012](https://doi.org/10.1016/j.eswa.2018.08.012).
- [47] S. Mirjalili, "SCA: A sine cosine algorithm for solving optimization problems," *Knowl.-Based Syst.*, vol. 96, pp. 120–133, Mar. 2016, doi: [10.1016/j.knsys.2015.12.022](https://doi.org/10.1016/j.knsys.2015.12.022).
- [48] W. Long, T. Wu, X. Liang, and S. Xu, "Solving high-dimensional global optimization problems using an improved sine cosine algorithm," *Expert Syst. Appl.*, vol. 123, pp. 108–126, Jun. 2019, doi: [10.1016/j.eswa.2018.11.032](https://doi.org/10.1016/j.eswa.2018.11.032).
- [49] S. Gupta and K. Deep, "Improved sine cosine algorithm with crossover scheme for global optimization," *Knowl.-Based Syst.*, vol. 165, pp. 374–406, Feb. 2019, doi: [10.1016/j.knsys.2018.12.008](https://doi.org/10.1016/j.knsys.2018.12.008).
- [50] C. Li, Z. Luo, Z. Song, F. Yang, J. Fan, and P. X. Liu, "An enhanced brain storm sine cosine algorithm for global optimization problems," *IEEE Access*, vol. 7, pp. 28211–28229, 2019, doi: [10.1109/ACCESS.2019.2900486](https://doi.org/10.1109/ACCESS.2019.2900486).
- [51] H. Nenavath, D. R. K. Jatoth, and D. S. Das, "A synergy of the sine-cosine algorithm and particle swarm optimizer for improved global optimization and object tracking," *Swarm Evol. Comput.*, vol. 43, pp. 1–30, Dec. 2017, doi: [10.1016/j.swevo.2018.02.011](https://doi.org/10.1016/j.swevo.2018.02.011).
- [52] X. Zhao, Y. Ji, and J. Hao, "A novel biogeography-based optimization algorithm with momentum migration and taxonomic mutation," in *Proc. Int. Conf. Swarm Intell.* Cham, Switzerland: Springer, 2020, pp. 83–93, doi: [10.1007/978-3-030-53956-6_8](https://doi.org/10.1007/978-3-030-53956-6_8).
- [53] X. Zhang, D. Wang, Z. Fu, S. Liu, W. Mao, G. Liu, Y. Jiang, and S. Li, "Novel biogeography-based optimization algorithm with hybrid migration and global-best Gaussian mutation," *Appl. Math. Model.*, vol. 86, pp. 74–91, Oct. 2020, doi: [10.1016/j.apm.2020.05.016](https://doi.org/10.1016/j.apm.2020.05.016).
- [54] X. Zhang, D. Wang, and H. Chen, "Improved biogeography-based optimization algorithm and its application to clustering optimization and medical image segmentation," *IEEE Access*, vol. 7, pp. 28810–28825, 2019, doi: [10.1109/ACCESS.2019.2901849](https://doi.org/10.1109/ACCESS.2019.2901849).
- [55] J. Cui, S. Wang, S. Wang, X. Zhang, S. Ma, and W. Gao, "Hybrid Laplace distribution-based low complexity rate-distortion optimized quantization," *IEEE Trans. Image Process.*, vol. 26, no. 8, pp. 3802–3816, Aug. 2017, doi: [10.1109/TIP.2017.2703112](https://doi.org/10.1109/TIP.2017.2703112).
- [56] P. Collard and J. P. Aurand, "DGA: An efficient genetic algorithm," in *Proc. ECAI*, 1994.
- [57] S. Yang and X. Yao, "Experimental study on population-based incremental learning algorithms for dynamic optimization problems," *Soft Comput.*, vol. 9, no. 11, pp. 815–834, Nov. 2005, doi: [10.1007/s00500-004-0422-3](https://doi.org/10.1007/s00500-004-0422-3).

- [58] J. Derrac, S. García, D. Molina, and F. Herrera, "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms," *Swarm Evol. Comput.*, vol. 1, no. 1, pp. 3–18, Mar. 2011, doi: [10.1016/j.swevo.2011.02.002](https://doi.org/10.1016/j.swevo.2011.02.002).
- [59] G. Xiong and D. Shi, "Hybrid biogeography-based optimization with brain storm optimization for non-convex dynamic economic dispatch with valve-point effects," *Energy*, vol. 157, pp. 424–435, May 2019, doi: [10.1016/j.energy.2018.05.180](https://doi.org/10.1016/j.energy.2018.05.180).
- [60] X. Sang, X. Liu, Z. Zhang, and L. Wang, "Improved biogeography-based optimization algorithm by hierarchical tissue-like p system with triggering ablation rules," *Math. Problems Eng.*, vol. 2021, pp. 1–24, Mar. 2021, doi: [10.1155/2021/6655614](https://doi.org/10.1155/2021/6655614).
- [61] S. Mahdavi, M. E. Shiri, and S. Rahnamayan, "Metaheuristics in large-scale global continues optimization: A survey," *Inf. Sci.*, vol. 295, pp. 407–428, Feb. 2015, doi: [10.1016/j.ins.2014.10.042](https://doi.org/10.1016/j.ins.2014.10.042).
- [62] Y. Ma and Y. Bai, "A multi-population differential evolution with best-random mutation strategy for large-scale global optimization," *Int. J. Speech Technol.*, vol. 50, no. 5, pp. 1510–1526, May 2020, doi: [10.1007/s10489-019-01613-2](https://doi.org/10.1007/s10489-019-01613-2).
- [63] W. Li, Z. Lei, J. Yuan, H. Luo, and Q. Xu, "Enhancing the competitive swarm optimizer with covariance matrix adaptation for large scale optimization," *Int. J. Speech Technol.*, vol. 51, no. 7, pp. 4984–5006, Jul. 2021, doi: [10.1007/s10489-020-02078-4](https://doi.org/10.1007/s10489-020-02078-4).
- [64] N. Awad et al., "Problem definitions and evaluation criteria for the CEC 2017 special session and competition on single objective bound constrained real-parameter numerical optimization," Nanyang Technol. Univ., Singapore, Tech. Rep., 2017.
- [65] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Adv. Eng. Softw.*, vol. 95, pp. 51–67, May 2016, doi: [10.1016/j.advengsoft.2016.01.008](https://doi.org/10.1016/j.advengsoft.2016.01.008).
- [66] M. Jain, V. Singh, and A. Rani, "A novel nature-inspired algorithm for optimization: Squirrel search algorithm," *Swarm Evol. Comput.*, vol. 44, pp. 148–175, Feb. 2019, doi: [10.1016/j.swevo.2018.02.013](https://doi.org/10.1016/j.swevo.2018.02.013).
- [67] A. Sajwan and A. Yadav, "AEFA: Artificial electric field algorithm for global optimization," *Swarm Evol. Comput.*, vol. 48, pp. 93–108, Aug. 2019, doi: [10.1016/j.swevo.2019.03.013](https://doi.org/10.1016/j.swevo.2019.03.013).
- [68] 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, doi: [10.1007/s10489-020-01893-z](https://doi.org/10.1007/s10489-020-01893-z).
- [69] M. Khishe and M. R. Mosavi, "Chimp optimization algorithm," *Expert Syst. Appl.*, vol. 149, Jul. 2020, Art. no. 113338, doi: [10.1016/j.eswa.2020.113338](https://doi.org/10.1016/j.eswa.2020.113338).
- [70] A. Faramarzi, M. Heidarinejad, S. Mirjalili, and A. H. Gandomi, "Marine predators algorithm: A nature-inspired Metaheuristic," *Expert Syst. Appl.*, vol. 152, Aug. 2020, Art. no. 113377, doi: [10.1016/j.eswa.2020.113377](https://doi.org/10.1016/j.eswa.2020.113377).
- [71] F. A. Hashim, E. H. Houssein, M. S. Mabrouk, W. Al-Atabany, and S. Mirjalili, "Henry gas solubility optimization: A novel physics-based algorithm," *Future Gener. Comput. Syst.*, vol. 101, pp. 646–667, Dec. 2019, doi: [10.1016/j.future.2019.07.015](https://doi.org/10.1016/j.future.2019.07.015).
- [72] M. H. Nadimi-Shahraki, S. Taghian, and S. Mirjalili, "An improved grey wolf optimizer for solving engineering problems," *Expert Syst. Appl.*, vol. 166, Mar. 2021, Art. no. 113917, doi: [10.1016/j.eswa.2020.113917](https://doi.org/10.1016/j.eswa.2020.113917).
- [73] G. Dhiman, M. Garg, A. Nagar, V. Kumar, and M. Dehghani, "A novel algorithm for global optimization: Rat swarm optimizer," *J. Ambient Intell. Humanized Comput.*, vol. 12, pp. 8457–8482, Oct. 2020, doi: [10.1007/s12652-020-02580-0](https://doi.org/10.1007/s12652-020-02580-0).
- [74] B. Abdollahzadeh, F. S. Gharehchopogh, and S. Mirjalili, "African vultures optimization algorithm: A new nature-inspired metaheuristic algorithm for global optimization problems," *Comput. Ind. Eng.*, vol. 158, Aug. 2021, Art. no. 107408, doi: [10.1016/j.cie.2021.107408](https://doi.org/10.1016/j.cie.2021.107408).
- [75] F. A. Hashim, E. H. Houssein, K. Hussain, M. S. Mabrouk, and W. Al-Atabany, "Honey badger algorithm: New Metaheuristic algorithm for solving optimization problems," *Math. Comput. Simul.*, vol. 192, pp. 84–110, Feb. 2022, doi: [10.1016/j.matcom.2021.08.013](https://doi.org/10.1016/j.matcom.2021.08.013).



ZIYU ZHANG received the B.Sc. degree from the School of Mathematical Sciences, Sichuan Normal University, China, in 2020. He is currently pursuing the M.Sc. degree with North Minzu University. His research interests include artificial intelligence and global optimization algorithms.



YUELIN GAO received the B.Sc. degree from Yan'an University, Yan'an, China, in 1984, and the M.Sc. degree from the Dalian University of Technology, Dalian, China, in 1991, and the Ph.D. degree from the Xi'an Jiaotong University, Xi'an, China, in 2002.

He is currently a Full Professor and a Doctor's Supervisor with the College of North Minzu University, Yinchuan, China. He has published more than 150 Academic papers in important journals.

His current research interests include global optimization, evolutionary computing, optimization theory and method, and financial statistics.



WENLU ZUO received the B.Sc. degree from the School of Mathematics and Information Science, Chongqing Normal University, China, in 2019. She is currently pursuing the M.Sc. degree with North Minzu University. Her research interest includes intelligent optimization algorithms.

• • •