An Improved Biogeography-based Optimization with Hybrid Migration and Feedback Differential Evolution and its Performance Analysis

Ziyu Zhang
North Minzu University
Yuelin Gao (✉ gaoyuelin@263.net)
North Minzu University https://orcid.org/0000-0003-2021-2097
Jiahang Li
North Minzu University
Wenlu Zuo
North Minzu University

Research Article

Keywords: Biogeography-based optimization (BBO), Convex migration, Opposition-based learning, Differential evolution, Feedback mechanism, Sequence convergence model

Posted Date: November 15th, 2021

DOI: https://doi.org/10.21203/rs.3.rs-1064223/v1

An Improved Biogeography-based Optimization with Hybrid Migration and Feedback Differential Evolution and its Performance Analysis

Ziyu Zhang[a,b], Yuelin Gao[a,b*], Jiahang Li[a], Wenlu Zuo[a]

(a. School of Mathematics and Information Science，North Minzu University，Yin chuan 750021，China;

b. Ningxia Province Key Laboratory of Intelligent Information and Data Processing，Yin chuan 750021，China)

ABSTRACT: Biogeography-based optimization (BBO) is not suitable for solving high-dimensional or multi-modal problems. To improve the optimization efficiency of BBO, this study proposes a novel BBO variant, which is named ZGBBO. For the selection operator, an example learning method is designed to ensure inferior solution will not destroy the superior solution. For the migration opeartor, a convex migration is proposed to increase the convergence speed, and the probability of finding the optimal solution is increased by using opposition-based learning to generate opposite individuals. The mutation operator of BBO is deleted to eliminate the generation of poor solutions. A differential evolution with feedback mechanism is merged to improve the convergence accuracy of the algorithm for multi-modal and irregular problems. Meanwhile, the greedy selection is used to make the population always moves in the direction of a better area. Then, the global convergence of ZGBBO is proved with Markov model and sequence convergence model. Quantitative evaluations, compared with three self-variants, seven improved BBO variants and six state-of-the-art evolutionary algorithms, experimental results on 24 benchmark functions show that every improved strategy is indispensable, and the overall performance of ZGBBO is better. Besides, the complexity of ZGBBO is analyzed by comparing with BBO, and ZGBBO has less computation and lower complexity.

KEYWORDS: Biogeography-based optimization (BBO); Convex migration; Opposition-based learning; Differential evolution; Feedback mechanism; Sequence convergence model

1 Introduction

Traditional optimization algorithms, such as linear programming, quadratic programming and convex optimization, usually require problems to be convex. But in real life, people often encounter many problems are not convex. The commonly used dynamic programming method can find the optimal solution, but its computational load increases exponentially with the size of the problem, so it is not suitable for large-scale problems. In contrast, although the population-based evolutionary algorithm can not guarantee to find the optimal solution of the problem, it has no requirements on the nature and scale of the problem, so it is more extensive and applicable. In half a century since the advent of genetic algorithm (Holland 1975), people have proposed a variety of evolutionary algorithms based on biological behavior and natural phenomena. Such as ant colony algorithm (ACO) (Dorigo 1992), particle swarm optimization (PSO) (Kennedy and Eberhart 1995), differential evolution algorithm (DE) (Storn and Price 1995), immune algorithm (IA) (Dasgupta 1998), bacteria foraging optimization (BFO) (Passino 2002), artificial bee colony (ABC) (Teodorovic et al. 2006), harmony search (HS) (Kang et al. 2005), etc. The evolutionary algorithm in artificial intelligence and machine learning has been playing an important role. Many researchers have proposed solutions based on these evolutionary algorithms for various optimization problems, and described techniques for dealing with these problems. However, these evolutionary algorithms have the disadvantages of slow convergence and easy to fall into the local optimal solution. In recent years, with the rapid development of science and technology, the practical problems in life have higher and higher requirements for algorithms, and the convergence accuracy and convergence speed of evolutionary algorithms need to be improved urgently. Therefore, inspired by the theories of physics, chemistry, humanities and other disciplines, scholars from all over the world put forward a variety of new and more advanced swarm intelligence evolutionary algorithms with better performance (Sloss and Gustafson 2020).

In 2016, Mirjalili and Lewis (2016) from Griffith University in Australia proposed the whale optimization algorithm (WOA) inspired by the hunting behavior of humpback whales. When whales hunt, they move their positions according to certain established rules. The whale with the best position guides other whales, and each whale updates the position according to the position updating equation provided by the mathematical model. WOA is a new population-based evolutionary algorithm, which has the advantages of simple operation, few parameters to adjust and strong ability to jump out of local optimum. Later, inspired by the phototaxis and flight modes of moths,

Email Address: 2541604895@qq.com (Z. Zhang), gaoyuelin@263.net (Y. Gao), 362781380@qq.com (J. Li), 1084366704@qq.com (W. Zuo).

Mohamed et al. (2017) proposed moth swarm algorithm (MSA) by simulating the behavior of moths flying toward the moon at night in nature. Different from most intelligence algorithms, the whole moth swarm is divided into moths of different action types in the moth swarm algorithm model, which more truly reflects the cooperative behavior between the population and balances the detection and mining ability of the algorithm to a certain extent. MSA has the advantages of simple structure, low complexity, high precision and strong robustness. In 2018, Elsisi et al. (2018) proposed a innovative search algorithm, future search algorithm (FSA), by imitating human beings' yearning for a better life. The algorithm builds a mathematical model to simulate the best life between human and human (local search) and the best life in history (global search) to obtain the optimal solution. Compared with other algorithms, FSA has the advantages of strong exploration ability and difficulty in falling into local optimal value. In 2019, Dhiman and Kumar (2019) proposed a novel global optimization algorithm named seagull optimization algorithm (SOA) inspired by the population behavior of seagulls in nature. Similar to other intelligent optimization algorithms, SOA is also a population-based evolutionary algorithm, which simulates the migration and attack behavior of seagull population. At the same time, inspired by Coulomb's law in physics, Anita and Yadav (2019) proposed the meta-heuristic artificial electric field algorithm (AEFA). AEFA simulates the motion of charged particles in an electrostatic field and evolves it into a random search for optimal solutions. AEFA is an outstanding nonlinear optimization algorithm with high convergence precision and wide adaptive range. In addition to AEFA, other evolutionary algorithm based on physics laws is electrostatic discharge algorithm (ESDA), which is a new meta-heuristic optimization algorithm proposed by Houssem et al. (2019) inspired by electrostatic discharge events. ESDA assumes that the fitness of electronic devices is related to the search position, and uses the phenomenon of direct or indirect electrostatic discharge between individuals electronic devices to change the location of electronic devices through the movement of electronic devices with low fitness value to electronic devices with high fitness value to obtain the optimal spatial position, which is the optimal solution of the problem. Compared with traditional intelligence algorithms, ESDA is a swarm intelligence evolutionary algorithm with excellent robustness, easy implementation and simple algorithm principle. In 2020, Faramarzi et al. (2020) proposed a new meta-heuristic optimization algorithm, which is named marine predator algorithm (MPA). It is inspired by the survival of the fittest theory in the ocean, in which marine predators choose the best foraging strategy between the Levy walk and the Brown walk. MPA is a novel evolutionary algorithm with advanced and applicability. Sparrow search algorithm (SSA) is also a population-based intelligent evolutionary algorithm proposed in the same year by Xue and Shen (2020). SSA is proposed mainly inspired by the foraging behavior and backfeeding behavior of sparrows. The core idea is that finders with better fitness values will get food first in the search process, which is the solution to the problem. This algorithm is novel and has the advantages of strong searching ability and fast convergence.

The biogeography-based optimization (BBO) studied in this paper is a population-based intelligent evolutionary algorithm proposed by American scholar Simon in 2008 (Simon 2008). Simon studied the migration phenomenon between habitats in biogeography and simulated the migration and information exchange mechanism between habitats to establish the mathematical model for optimization, so as to obtain the intelligent optimization process. Compared with other evolutionary algorithms, BBO has fewer parameter setting and is easier to implement. BBO algorithm has good population information utilization ability, but BBO search ability is weak, prone to premature algorithm phenomenon, and the convergence rate in the late evolution is very slow (Guo et al. 2017). In order to enhance the search ability of the original BBO, improve its ability to jump out of the local optimal solution, and improve the convergence speed of the algorithm in the late stage, this paper proposes a novel biogeography-based optimization, referred to as ZGBBO. In the new algorithm, the mutation operator of the original BBO is deleted to reduce its computational complexity and eliminate the disadvantages caused by random mutation. Using the differential evolution with feedback mechanism to replace the original mutation operator, so the population can select the mutation mode intelligently according to the change of the current optimal value, which improve the optimization accuracy of the algorithm for multi-modal, irregular function. Secondly, the example learning method is used to replace the roulette selection operator of the original BBO, so that the poor solution will not have adverse effects on the better solution, and the algorithm reduces the calculation of habitat emigration rate, which further reduces the calculation amount. Thirdly, in view of the direct migration mode in which variables of emigration individual directly replace variables of immigration individual in the original BBO, convex migration mechanism is adopted in this paper to replace it, so that the immigration individual will definitely accept some variables from the optimal solution. When the lower ranked individuals in the population are not selected for migration, opposition-based learning is used to update their positions. The opposite individuals with poor solutions are generated to improve the probability of finding the optimal solution of the problem. Finally, the greedy selection strategy is adopted to ensure that the optimal individual in each generation is not inferior to the optimal solution of the previous generation, so the population always moves towards the direction of a better solution.

The ZGBBO algorithm proposed in this paper can not only effectively use the information utilization ability of original BBO, but also effectively combine the search ability of DE to achieve the balance between information utilization and search. We compare the performance of ZGBBO with its three variant algorithms, seven improved BBO algorithms and other six state-of-the-art evolutionary algorithms, and find that ZGBBO has better overall performance, higher convergence accuracy and faster convergence rate among all comparison algorithms. At the same time, we use two methods to prove the convergence of ZGBBO algorithm. On the one hand, Markov model is used to prove the convergence of ZGBBO algorithm. On the other hand, a new method is proposed by establishing a sequence convergence model to prove that ZGBBO has global convergence. In addition, this research also analyzes the algorithm complexity of ZGBBO. By comparing the time consumption and computational amount of the original BBO, it can be seen that ZGBBO is an evolutionary algorithm with less computational amount and less time consumption.

The graphical abstract of this paper is shown in Fig. 1.

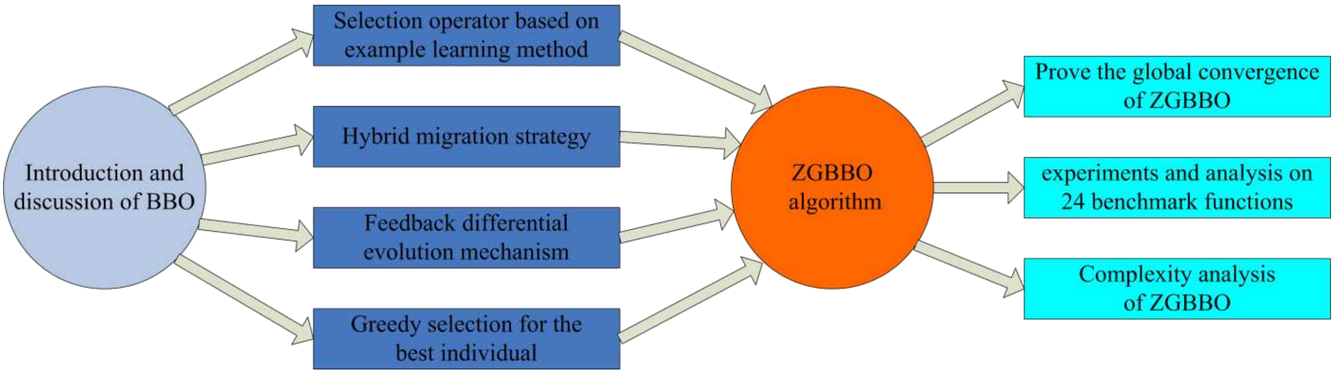

Introduction and discussion of BBO
Selection operator based on example learning method
Hybrid migration strategy
Feedback differential evolution mechanism
Greedy selection for the best individual
ZGBBO algorithm
Prove the global convergence of ZGBBO
experiments and analysis on 24 benchmark functions
Complexity analysis of ZGBBO

Fig. 1 Graphical abstract of this paper

The rest of this paper is summarized as follows: Chapter 2 mainly introduces the standard biogeography optimization algorithm, including the calculation steps, related work and main shortcomings of BBO. Chapter 3 mainly describes the improvement strategy of BBO in this paper and proposes the improved algorithm ZGBBO. In Chapter 4, the global convergence of ZGBBO is proved in detail. Chapter 5 is a numerical experiment in which the performance of ZGBBO is compared with other algorithms and the results are analyzed Chapter 6 analyzes the complexity of the algorithm in this paper and compares it with the time consumption and calculation amount of the original BBO. Finally, the thesis is summarized in Chapter 7, and the future research direction is pointed out.

2 Discussion of BBO

Biogeography studies the formation, extinction and geographical distribution of biological species (Simon 2008). Therefore, the mathematical model of biogeography-based optimization mainly describes species formation, migration between islands, and extinction. An "island" here refers to a habitat that is geographically isolated from other habitats. An island that is suitable for living organisms can be interpreted as having a high habitat suitability index (HSI). Factors that influence whether an island is suitable for life include rainfall, surface temperature, plant diversity, land area, and topographic features et al., which can be understood as features associated with HSI. These variables determine whether an island is habitable and are called suitability index variables (SIVs). Biogeography-based optimization algorithm is a relatively new evolutionary algorithm, which will be discussed systematically in this chapter.

2.1 Standard BBO

Suppose there is a global optimization problem and some candidate solutions, which are called individuals. Good individuals perform better on the problem, that is, the function value obtained is better; and bad individuals perform poorly on the problem, so the function value is worse. Good individuals are like islands with high HSI, and bad individuals are like islands with low HSI. Good individuals are more likely to share their features (i.e. independent variables of candidate solutions) with poor individuals, and poor individuals are more likely to accept the features of good individuals. The addition of new features may improve the quality of poor individuals, so that the habitat can be improved and the target function value can be better, which is the core idea of BBO. The correspondence between BBO algorithm and biogeography theory is listed in Table 1.

Table 1 Correspondence between BBO and biogeography theory

Biogeography theory	**BBO algorithm**
Island	Individual (candidate solution)
Habitat suitability index (HSI)	Objective function value
Suitability index variables (SIVs)	Independent variables of candidate solutions
Catastrophic events transformed the	Mutation
The number of islands	Population size (number of candidate solutions)
Islands with low HSI tend to immigrate	Poor individuals tend to accept variables from good
Islands with high HSI tend to emigrate	Good individuals tend to share variables with poor

BBO mainly improves the quality of habitats through population evolution, and through migration between species to find the most suitable habitat for the survival of species, which is the optimal solution of the optimization problem. It is mainly accomplished by the following three steps.

2.1.1 Initialization

The initialization steps of BBO include parameter initialization and population initialization. Parameter initialization includes setting the population size *NP*, the maximum number of species that can be accommodated in each habitat $S_{\max}$, the maximum immigration rate *I* and the maximum emigration rate *E*. Only when $S_{\max}$、 *I* and *E* are determined, can we calculate the immigration rate λ_i、 emigration rate μ_i、 species number S_i and species probability P_i for each habitat. In addition, in order to calculate the mutation rate m_i of each habitat, the maximum mutation rate $m_{\max}$ of the habitat should be set.

When initializing the population, *NP* habitats are randomly generated as the original population. The operation is that each habitat randomly generates a number within a range of values on each variable. Assuming that each habitat has *n* variables, each habitat is initialized according to Eq. (1).

$$x_i(k) = rand(0,1)\cdot(x_{k\max} - x_{k\min}) + x_{k\min} \tag{1}$$

Where $i \in \{1,2,\cdots,NP\}$, $k \in \{1,2,\cdots,n\}$. $x_i(k)$ is the *k*-th variable of habitat x_i ,while $x_{k\min}$ and $x_{k\max}$ are the lower and upper bounds of the *k*-th variable of each habitat. $rand(0,1)$ represents a number in the range of (0,1) generated randomly to ensure that the variable does not exceed the bound. Once the population is initialized, the HSI, which is the evaluation function value, for each habitat can be calculated to measure the quality of each habitat.

2.1.2 Migration

BBO algorithm designs migration operators between habitats, and uses the migration rate of each individual to share information between individuals according to probability (Simon 2008). Therefore, there is a positive correlation between habitat HSI and species number *S*. This means that the higher the HSI, the greater the *S*, and vice versa. After the fitness values of *NP* habitats were calculated, each x_i was sorted in descending order, that is, x_i was rearranged according to HSI from high to low. Eq. (2) provides the calculation method of species quantity in the sorted habitats.

$$S_i = S_{\max} - i,\ \ i \in \{1,2,\cdots,NP\}. \tag{2}$$

According to the number of species in the habitat, the immigration rate and emigration rate of each habitat can be calculated. Different migration rate models will have an important impact on the optimization performance of the algorithm. Ma (2010) proved through experiments that the performance of complex migration rate model is better than that of simple migration rate model. Therefore, when BBO algorithm and other improved BBO algorithms are implemented in this paper, the cosine migration rate model proposed by Ma. is used to calculate the immigration rate and emigration rate of each habitat, as shown in Eq. (3).

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

Where, *I* is the maximum immigration rate, *E* is the maximum emigration rate, and $S_{\max}$ is the maximum number of species.

BBO will first determine the feature *k* of habitat x_i to be migrated according to the immigration rate λ_i when performing the migration operation. In each generation, the probability of substitution for each independent variable of the *i*-th individual is λ_i, that is, λ_i is equal to the probability of substitution for each independent variable of x_i. $rand(0,1)$ is used to generate random numbers between (0,1). If the random number is smaller than the immigration rate, this dimension feature of habitat x_i need to be replaced, and

x_i is also known as the habitat to be immigrated. If we have choosed the variable of the solution to be replaced, then we choose to emigrate individual x_j with a probability proportional to the emigration rate $\{\mu_i\}$. According to Eq. (4), the original BBO uses the "roulette" method to determine the habitat x_j to be emigrated, and finally replaces the k-dimension variable of x_i with the k-dimension variable of x_j. Algorithm 1 gives the calculation flow of BBO migration operator.

$$P(x_j \text{ is selected})=\frac{\mu_j}{\sum_{i=1}^{N}\mu_i}. \tag{4}$$

Algorithm 1 BBO migration operator

```
for each habitat $x_i$
    for each dimension variable $k$
        if rand(0,1)< $\lambda_i$
            according to $\{\mu_i\}_{i=1}^{NP}$, the habitat $x_j$ to be emigrated is selected by roulette
            $x_i(k) = x_j(k)$
        end if
    end for
end for
```

Fig. 2 is an example of a BBO algorithm migration. Where, z_i is the immigration habitat, x_i is the emigration habitat. From the example in Fig. 2, you can see the following migration decision:

(1). The first dimension feature is selected for immigration, and x_2 is selected as the emigration individual according to Eq. (4). So the first dimension feature of z_i is replaced by the first dimension feature of x_2.

(2). The second dimension feature is selected for immigration, and x_1 is selected as the emigration habitat according to Eq. (4). So the second dimension feature of z_i is replaced by the first dimension feature of x_1.

(3). The third dimensional feature are not selected for immigration, so the third dimensional feature of z_i remain unchanged.

(4). The fourth dimension feature is selected for immigration, and x_{NP} is selected as the emigration habitat according to Eq. (4). So the fourth dimension feature of z_i is replaced by the fourth dimension feature of x_{NP}.

(5). The fifth dimension feature is selected for immigration, and x_3 is selected as the emigration individual according to Eq. (4). So the fifth dimension feature of z_i is replaced by the fifth dimension feature of x_3.

(6). The sixth dimension feature is selected for immigration, and x_2 is selected as the emigration habitat according to Eq. (4). So the sixth dimension feature of z_i is replaced by the sixth dimension feature of x_2.

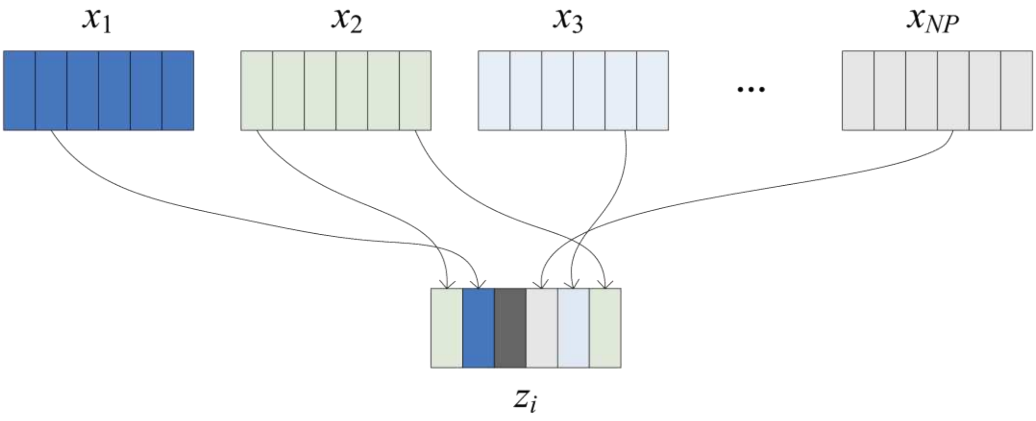


Fig. 2 BBO migration example for 6-dimensional problems

2.1.3 Mutation

In biogeographical theory, HSI in each habitat may undergo drastic changes due to sudden disease or natural disaster of species (Ma et al. 2017). Volcanic eruptions, for example, can strip islands of any species but make them more habitable, making the environment more hospitable to new immigrations. Therefore, the sharp change of HSI can make the habitat more livable or change the habitat from an environment suitable for living organisms to an environment not suitable for living organisms. In the standard BBO algorithm, the species probability is calculated by Eq. (5) according to the immigration rate and emigration rate of the habitats.

$$P_i = \begin{cases} -(\lambda_i+\mu_i)\ P_i + \mu_{i+1}P_{i+1}, & S_i=0 \\ -(\lambda_i+\mu_i)\ P_i + \lambda_{i-1}P_{i-1} + \mu_{i+1}P_{i+1}, & 1 \le S_i \le S_{\max}-1. \\ -(\lambda_i+\mu_i)\ P_i + \lambda_{i-1}P_{i-1}, & S_i = S_{\max} \end{cases} \tag{5}$$

The species probability of a habitat is inversely proportional to the mutation rate in that habitat. Habitats with high species probability had better environment and less mutation probability. Habitats with low species probability are poorer, less suitable for species and more likely to undergo environmental mutations. In the original BBO, Eq. (6) was used to calculate the habitat mutation rate m_i.

$$m_i = (1-\frac{P_i}{P_{\max}})\cdot m_{\max},\ \ P_{\max} = \max(\{P_i\}_{i=1}^{NP}). \tag{6}$$

Where, $m_{\max}$ is the maximum mutation rate and is given as the initial value. Like the immigration rate and the emigration rate, the mutation rate m_i of habitat x_i is equal to the probability of mutation for each variable in x_i. During mutation, a random number between (0,1) is generated for each habitat. If the random number is less than the mutation rate m_i, a number within the value range is randomly generated for each dimension variable to replace the original variable value. Algorithm 2 gives the calculation flow of BBO mutation operator.

Algorithm 2 BBO mutation operator

```
for each habitat $x_i$
    if rand(0,1)< $m_i$
        for each dimension variable $k$
            $x_i(k) = rand(0,1)\cdot(x_{k\max} - x_{k\min}) + x_{k\min}$
        end for
    end if
end for
```

BBO algorithm abstracts the migration and mutation of species in biogeography by constructing a mathematical model to describe the dynamic distribution of species in biogeography. BBO is a new evolutionary algorithm with good utilization of population information. Now, we describe the computational pseudocode of the BBO algorithm in detail in Algorithm 3.

Algorithm 3 BBO calculation process

```
initialization parameters, including $NP$, $S_{\max}$, $m_{\max}$, etc
use Eq. (1) to generate the initial population
while (the termination condition is not met)
    calculate the fitness HSI for each habitat
    rank all habitats from highest to lowest in terms of suitability
    use Eq. (2) to calculate the species number $S_i$ in each habitat
    use Eq. (3) to calculate the immigration rate $\lambda_i$ and emigration rate $\mu_i$ of each habitat
    use Equations (5) and (6) to calculate the mutation rate $m_i$ of each habitat
    perform Algorithm1 to complete the migration operation
    perform Algorithm2 to complete the mutation operation
end while
```

2.2 The main disadvantages of BBO

There are some defects in the three main operators (selection, migration and mutation) of BBO, and the convergence speed of the algorithm is slow in the late stage, and it is easy to fall into the local optimal solution, which affects the optimization performance and running speed of the algorithm. The reasons for these shortcomings of BBO will be analyzed in depth below.

（1） It can be seen from Eq. (4) that the original BBO algorithm uses roulette to select habitats for emigration. Although the probability of being selected is proportional to the emigration rate of habitats, it cannot avoid the immigration of inferior individuals to superior ones. For each habitat, if selected for immigration, the corresponding emigration habitat is selected from the remaining *NP*-1 habitats. If habitat x_i is immigrated, it is likely to be emigrated of habitat x_j (j>i), which means that habitats with lower HSI immigrate to habitats with higher HSI, and variable values of poorer individuals replace variable values of better individuals, thus reducing fitness of better individuals and causing "negative action". Therefore, the random selection method based on roulette tends to bring in poor individuals, reduce the population quality and slow down the convergence rate.

（2） The original BBO uses the method of direct migration to replicate the variables between candidate solutions, and the migration operator will directly replace the corresponding features of the immigration habitat with the features of the emigration habitat. However, if an individual performs well on the questions, it does not mean that the individual performs well on the variable values of each dimension. Therefore, the original migration operator is completely possible to make a dimension of the habitat become worse after being replaced, and the suitability is reduced. Moreover, the single migration method is not suitable for solving the multi-modal problem, and it is easy to make the algorithm fall into the local optimal value, resulting in search stagnation.

（3） The original BBO uses random mutation to improve population diversity and help the algorithm jump out of the local optimal solution. For individuals with low fitness, random mutation tends to produce better individuals and improve the quality of the population. However, for individuals with high fitness, random mutation can easily destroy them, leading to worse individuals and lower population diversity. This way of mutation is blind and cannot guarantee the mutation to the direction of the optimal solution. In addition, the mutation operator needs to calculate the species probability and mutation rate of all habitats. From Eq. (5), it can be seen that the calculation of species probability is complicated and requires a lot of calculation, which consumes resources. Therefore, the random mutation strategy is not only easy to destroy the individuals with high fitness, but also increases the time consumption of the algorithm and reduces the convergence speed of the algorithm.

（4） In the biogeography-based optimization, the information transfer mechanism between individuals makes it have good utilization ability of population information, and can efficiently utilize existing habitat information for optimization. However, the BBO search ability is weak. It only relies on the substitution of several variables to search the problem space, so there is less chance to generate new solutions. Due to the lack of BBO search ability, the population diversity of BBO decreases rapidly in the late iteration, and the convergence rate of the algorithm is slow in the late evolution. Improving BBO's mining capacity has also been a research focus.

2.3 Review of BBO's work

Since BBO was proposed, it has been widely favored by scholars around the world. In the first year after the algorithm was proposed, 38 articles about BBO were published, 81 in the second year, and 145 in the third year. In recent years, the BBO algorithm is still emerging in an endless stream of articles. In order to improve the optimization performance of BBO algorithm, many scholars have conducted a lot of research. We will briefly review the research work on BBO in the past five years.

Loon et al. (2016) integrated tabu search (TS) into BBO in 2016, replacing the original mutation operator with the tabu search process. Experimental results show that this hybrid algorithm (BBOTS) can overcome the deficiency caused by random mutation and find the ideal solution in a reasonable time. Later, Yogesh et al. (2016) developed the PSOBBO algorithm, which greatly improved the computing performance of the original BBO. In order to improve the slow speed of BBO in global search and improve the detection ability of BBO, PSOBBO integrated particle swarm optimization algorithm into BBO and designed a new particle speed and position update method. This algorithm improves the half of the particles with poor fitness so that the particles are easily attracted to the global optimal position. Liu et al. (2018) then introduced NEH algorithm into BBO and obtained IBBO algorithm with good population quality and diversity. In order to improve the BBO algorithm, a migration operator based on insertion rules and a mutation operator based on exchange rules are proposed, and an neighborhood search algorithm is adopted to enhance the local search capability of IBBO. Then, Mehta et al. (2018) redesigned the calculation of mutation rate for each habitat to reduce the possibility of mutation for excellent individuals (MBBO). MBBO improves the migration operator by introducing the exponential average of the optimal solution, and updates the habitat location by using the exponential average of the variable values of the existing habitat and the variable values of the selected habitat. In 2019, with the proposal of many new intelligent algorithms, Pushpa Farswan et al. (2019) integrated fireworks algorithm (FWA) (Tan et al. 2010) into biogeography-based optimization,and named this hybrid algorithm as FBBO. The main work of FBBO is to mix two different search techniques to improve the quality of the solution and achieve a balance between the information utilization and the search ability of the algorithm. Then Reihanian et al. (2019) designed NBBO algorithm to overcome the difficulty that BBO could not effectively balance its development ability and search ability. Under the framework of NBBO algorithm, only two or more sub-iterations are considered in each iteration to perform the search task. In each sub-iteration, a sub-population is selected from the current population according to the triangular probability distribution, and the migration habitat is selected from this sub-population. In addition, this algorithm also introduces a two-stage migration operator into BBO, which enables the algorithm to search for the optimal solution quickly. The overall framework of NBBO makes its development ability and search ability reach a good balance, and it is not easy to fall into the local optimal state. Moreover, the mutation operator of the original algorithm is deleted, which makes up for the calculation amount brought by the new strategy. In order to reduce the computational complexity, Zhang et al. (2019a) also deleted the mutation operator in BBO, combined the differential mutation and sharing operator into BBO's migration operator, and combined the improved migration operator with one-dimensional and full-dimensional search strategies to conduct alternate search. Because many operators are integrated into the new algorithm to improve the search performance, it is named efficient and merged biogeography-based optimization (EMBBO). The same year, in order to alleviate the rotation variance of BBO and overcome its premature convergence problem, TDBBO was designed by Zhao et al. (2019). TDBBO adopts linear migration model and sinusoidal migration model respectively in the early and late evolution stage, and uses differential mutation operator to alleviate rotation variance. TDBBO is much better than the original BBO algorithm in solution quality, convergence speed and stability. In 2020, An et al. (2020) proposed an improved non-dominated sequencing biogeography-based optimization (INSBBO) to solve the multi-objective flexible job-shop scheduling problem. In order to overcome the pressure of individual selection in pareto dominance principle, INSBBO proposed a V-dominance algorithm based on the volume surrounded by the value of normalized objective function to enhance the convergence of the algorithm. On the other hand, in order to avoid the loss of some better solutions in the process of evolution, the author constructed an elite storage strategy (ESS) to store these better solutions, and improved the migration and mutation operators of the original BBO, which further improved the optimization ability of the algorithm. Meanwhile, BBO was also designed as a binary algorithm for feature selection (FS) (Albashish et al. 2020). The support vector machine recursive feature elimination (SVM-RFE) is embedded into the BBO to improve the quality of the obtained solutions in the mutation operator which striking an adequate balance between exploitation and exploration of the original BBO. The new method, BBO-SVM-RFE, outperforms the BBO method and other existing wrapper and filter methods in terms of accuracy and number of selected features. In 2021, Ghatte (2021) studied and discussed firefly algorithm (FA) and BBO, and fused the two algorithms to obtain a hybrid algorithm (FABBO). One defect of FA is that all the individuals will converge to a better solution at the end of iteration, so the algorithm will converge to a local optimal, which is not suitable for global optimization problems. Considering that the global search capability of FA is weak and the local search capability of BBO is weaker than FA, so the integration of FA and BBO can overcome the above defects. FABBO is essentially a two-stage algorithm: in the first stage, FA is used for preliminary optimization to find some local optimal solutions; The second stage uses BBO to perform a more refined search of the solution obtained in the first stage. Recently, Sang et al. (2021) proposed an improved BBO algorithm by hierarchical tissue-like P system with triggering ablation rules in view of many shortcomings of BBO in terms of global optimization, convergence speed and algorithm complexity, which is named DCGBBO. Sang et al. first proposed a dynamic crossover migration operator to improve the global search capability and increase species diversity. Then, dynamic Gaussian mutation operator is introduced to accelerate the convergence speed and improve the local search ability of the algorithm. Finally, the hierarchical tissue-like P system is combined with BBO to implement the migration and mutation of habitats by using evolutionary rules and communication rules, which fully reduces the computational complexity.

The research on the BBO has been continuously since the algorithm was put forward, and it has always been the hot spot of scholars in various countries. In just 13 years, BBO has developed theoretical materials about it, including Markov model, dynamic system model, statistical mechanics model, etc. (Ma et al. 2017). Although the improvement of BBO by many scholars has improved the performance of the original algorithm to a certain extent, with the rapid development of modern science and technology, practical problems in life have higher and higher requirements on algorithm performance, so the overall performance of BBO still need to be improved.

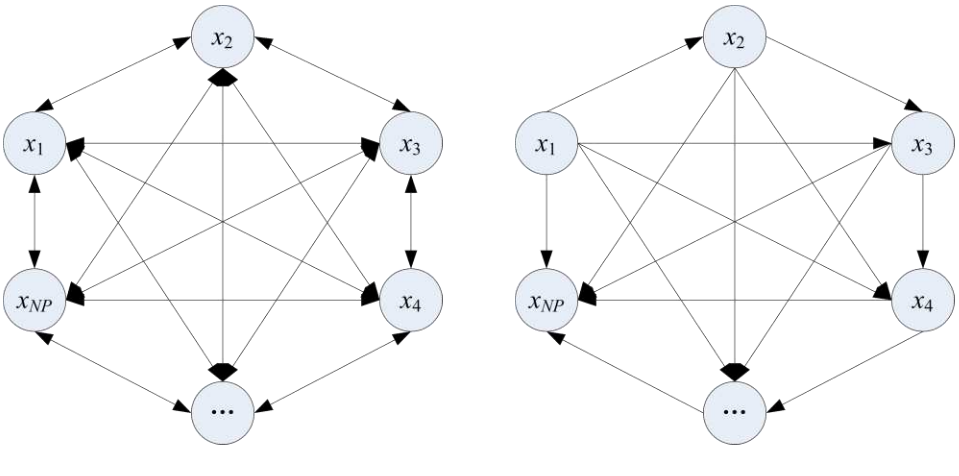
3 ZGBBO algorithm

In section 2.2, we analyze the shortcomings of the standard biogeography-based optimization in detail. In order to improve the algorithm performance and make BBO algorithm more advanced and superior, this paper proposes four improvement strategies for the four defects of BBO in section 2.2 through a large number of experiments and studies, which effectively make up for the shortcomings of the original BBO. The new algorithm is a highly competitive improved algorithm with fast convergence speed and high convergence precision when searching the optimal solution. We named it ZGBBO after the surnames of the first two authors of this paper. The algorithm principle and calculation process of ZGBBO will be described in detail in this chapter.

3.1 Selection operator based on example learning method

In view of the first deficiency of the original BBO in section 2.2, we use example learning method to replace the original roulette wheel selection method. Random selection of roulette tends to bring in poor individuals, which reduce the population quality and slow down the convergence rate. The ranking of each individual in a population is related to its HSI value, which means the higher the HSI, the higher the habitat ranking, and the lower the HSI, the lower the ranking. Therefore, in the improvement strategy, we set examples based on the ranking of each habitat and proposed the example learning method. For habitat x_i, it ranks *i* out of all habitats, and the HSI of other habitats higher than habitat x_i can only be x_1, $x_2, \cdots$, x_{i-1}. These habitats rank higher than x_i, so they become the role models of x_i, and x_i becomes the learner. In order to explain the principle of example learning method more intuitively, we will elaborate in detail according to Fig. 3.

Assuming that all individuals in the population have been ranked, x_1 is the best individual and x_{NP} is the worst individual. As shown in Fig. 3 (a), it is a random selection mode. Each of the two habitats can migrate with each other. x_1 can be replaced by any lower quality solution, while x_{NP} can migrate variable values to any better solution. Therefore, the random selection operator will cause the bad solution to destroy the better solution, thus reducing the population diversity of the algorithm. Fig. 3 (b) is example learning method. It can be observed that only one-way migration can be carried out between individuals, that is, the poor solution can only accept variable values from the better solution, and the poor solution cannot affect the better solution. For example, only x_1 is ranked higher than x_2, so x_2 can only accept features from x_1, while individuals ranked lower than x_2 can accept features from x_2, but cannot migrate to x_2. For an individual, those who are better than itself are role models, while those who rank lower than itself are learners. Therefore, x_1 can only be a role model, not a learner. x_{NP} can only be a learner, not a role model.



(a) Random selection

(b) Example learning

Fig. 3 Schematic diagram of random selection operator and example learning method

During the migration of x_i, a habitat from ($1, i$ -1) is randomly selected as a role model for migration. The role model x_j of x_i can be selected by Eq. (7):

$$j = round(unifrnd(1, i-1)). \tag{7}$$

unifrnd(*) is a random number between 1 and *i*-1,while *round*(*) is an integer function that computes the ranking of sample individuals. In section 2.3, all the work on BBO uses random selection operators, so none of these algorithms avoids the bad influence of poor solutions on better ones. The comparison algorithm in this paper, PRBBO (Feng et al. 2017), takes this problem into account and adopts a selection method based on random topological rings, so that the habitat can only migrate between adjacent individuals. However, this operation only reduces the probability of the inferior solution replacing the superior one, and does not eliminate this phenomenon. Moreover, the loop topology reduces the algorithm's ability to utilize population information, which makes the algorithm converge slowly. Using the example learning method to select the emigrating habitat can ensure that the migrating habitat to x_i has a higher fitness than x_i, while the better solution cannot be replaced by the poor solution, avoiding the situation that the good solution is destroyed by the poor solution. In addition, the example learning method does not need to calculate the emigration rate of each habitat, which reduces one calculation step and reduces the calculation amount.

3.2 Hybrid migration strategy

Section 2.2 points out that the second disadvantage of the biogeography-based optimization is the use of direct migration operators. This single migration method is not suitable for solving the multi-modal problem, and it is easy to make the algorithm fall into the local optimal value, resulting in search stagnation. Aiming at the above shortcomings, we integrate convex migration mechanism and opposition-based learning strategy into BBO, and obtain a hybrid migration operator. Firstly, the convex migration mechanism is used to replace the original migration operator. In convex migration, the variable of immigration habitat is no longer a duplicate of the variable of emigration, but is replaced by a convex combination of the variables of the two habitats. In migration, the emigration habitat is selected by example learning method in section 2.1, and the other paternal gene is from the optimal individual of the current population. The specific calculation equation is shown in Eq. (8) :

$$x_i(k) = \alpha \cdot x_{best}^t(k) + (1-\alpha) \cdot x_j(k). \tag{8}$$

Where $\alpha \in rand(0,1)$, x_{best}^t is the current optimal individual, t is the current iteration number, and x_j is the example selected through Eq. (7). It can be seen that when α=0, Eq. (8) degenerates into the original migration operator of BBO. Convex migration is suitable for continuous solution characteristic problems and can also be adjusted for discrete solution characteristic problems. Compared with the original migration operator, there are three advantages of migration using Eq. (8): (1) The good individuals are less likely to degenerate due to migration, because part of their original characteristics will be retained in the process of migration. (2) The poor individuals will accept the characteristics of the solution from at least part of the optimal individual in the migration. (3) Such migration ensures that the population evolves towards the direction of the optimal value of each generation, no longer blindly searches, and can quickly converge towards the direction of the optimal solution.

Second, the opposition-based learning strategy is integrated into the original BBO. Convex migration mechanism can only ensure the selected individuals to move towards the optimal solution direction, but can not help the rest of the individuals to obtain a better position. Therefore, it is necessary to design the updating formula for the position of the remaining individuals. When the random number generated on a variable of habitat x_i is greater than the immigration rate λ_i, this dimensional variable still needs to be replaced. The specific operation is to generate the opposite individual $\overline{x_i}$ of habitat x_i, and replace the variable of habitat x_i with the variable of opposite individual $\overline{x_i}$. Since Ergezer et al. (2014) proved that the quasi-reflective opposite individual has a large probability to approach the optimal solution of the problem. The opposite individual $\overline{x_i}$ of x_i is generated by Eq. (9):

$$\overline{x_i}(k) = unifrnd(x_i(k), \frac{x_{k\min} + x_{k\max}}{2}), k = (1,2,\cdots,D). \tag{9}$$

It can be seen from Eq. (9) that the variable value of each dimension of the opposite individual $\overline{x_i}$ is equal to the random number between the variable value of each dimension of x_i and the median value of the upper and lower bounds of the variable of this dimension. However, implementing the opposition-based learning strategy requires a lot of computing resources, and an additional fitness evaluation is required for each generated opposite individual. When performing evolutionary algorithms, opposite individuals cannot be generated randomly, only when there is reason to believe that additional calculations will lead to better results. Although quasi-reflective opposite is used in literature (Ergezer et al. 2014) to generate opposite individuals, it carries out opposition-based learning for all individuals in the population. In fact, the top individual is not worth generating its opposite individual. EMBBO (Zhang et al. 2019a) generates its opposite individual for only one individual in the population, which fully reduces the complexity of the algorithm. But this individual is not the last one, it is chosen at random. Therefore, it is possible to generate opposite solutions of the better solutions, thus reducing the diversity of the population. Therefore, when the opposition-based learning strategy is applied to the BBO in this paper, only *NP*/2 individuals from the less adaptable part of the population are taken to generate opposite individuals. A well-adapted individual is unlikely to produce an opposite individual that is stronger than the original. In other words, the solution closer to the optimal value is not worth generating its opposite solution. Random generation of opposite individuals will not only waste the evaluation times of the function, but also reduce the quality of the population, so only the opposite solution of poor individuals will be generated.

Combined with the example learning method in section 3.1, convex migration mechanism and opposition-based learning strategy in section 3.2, the pseudo-code of migration operator of ZGBBO is given in Algorithm 4.

Algorithm 4 ZGBBO migration operator

```
for each habitat $x_i$
    for each dimension variable $k$
        if rand(0,1)< $\lambda_i$
            Eq. (7) is used to select example habitat $x_j$
            use Eq. (8) for convex migration
        else
            use Eq. (9) to generate the opposite individual $\overline{x_i}$ of $x_i$
            $x_i(k) = \overline{x}_i(k)$
        end if
    end for
end for
```

3.3 Feedback differential evolution mechanism

The improvement strategies in Sections 3.1 and 3.2 are mainly used to improve the search efficiency of the algorithm, but cannot effectively help the algorithm jump out of the local optimal solution. The mutation operator can add new individuals to the population, so the algorithm is not easy to fall into the local optimal state. However, as can be seen from section 2.2, the random mutation method of the original BBO is easy to generate low-quality individuals and reduces population diversity. Random mutation can not effectively help the algorithm to jump out of the local optimal solution, and the calculation of species probability consumes CPU resources. Therefore, in the calculation process of ZGBBO, we delete the random mutation operator, which further reduces the computational complexity and eliminates the bad solution caused by random mutation. Meanwhile, in view of the weakness of BBO's weak search ability pointed out in section 2.2, this paper integrates the differential evolution algorithm into BBO, and uses DE's search ability to balance the information utilization capacity and new solution development capacity of BBO. Therefore, we design differential evolution with feedback mechanism in ZGBBO to replace the original mutation operator, so that the population can select the mutation mode intelligently according to the change of the current optimal value.

In the differential evolution algorithm, each generation of population adds new genetic information to the individuals through the addition, subtraction and scaling operation between multiple different vectors to generate mutant individuals, so as to realize the population replacement. Mutation operation is a key step to help the algorithm jump out of local optimal state and search for optimal value. The original differential evolution algorithm has a variety of mutation mechanisms. Through comparative analysis, the characteristics and suitable situations of various mutation formulas are listed in Table 2.

The vector mutation mode of the traditional differential evolution algorithm is always set in advance and will not be affected by the change of the overall fitness value of the population. In other words, the whole differential evolution system is a "static" evolution process. However, a single mutation method cannot be suitable for all objective functions and optimization problems, and a single mutation mechanism will lead to low adaptability and small scope of application of the algorithm. In addition, different mutation modes are applicable to different search states, and the population cannot adjust its search direction in time according to individual distribution in the search space during evolution. For example, when an individual falls into the local optimal solution, it is very likely to bring the whole population into the local optimal state, so the population diversity decreases. At this time, random individuals are needed to help the population jump out of the local optimal solution. In this case, the mutation approach is suitable for using DE/ rand /1 and DE/ rand /2. For a large search range, the population is easy to be distributed in each region of each space during initialization, so the search directions are different, and the algorithm is easy to diverge. At this time, it is necessary to make the population move towards the direction of the optimal individual to improve the search efficiency and the convergence speed of the algorithm. In this case, mutation approaches DE/ best /1 and DE/ best /2 are optimal choices. Any single mutation method can not adjust the search direction of the population in time during the evolution process, which reduces the flexibility and convergence accuracy of the algorithm.

Table 2 Mutation strategies of differential evolution algorithm

Differential type	Mutation formula	Characteristics	Suitable situation
DE/rand/1	$v_i = x_{r1} + F(x_{r2} - x_{r3})$	High population diversity is maintained, but the convergence rate is slow	Multimodal function
DE/best/1	$v_i = x_{best}^t + F(x_{r1} - x_{r2})$	The population as a whole moves towards the optimal solution with fast convergence speed, but it is easy to fall into the local optimal solution	Unimodal function
DE/rand-to-best/1	$v_i = x_i + F(x_{best}^t - x_i + x_{r1} - x_{r2})$	Combined with the characteristics of random model and optimal model, it is beneficial to balance the convergence speed and accuracy, but it is difficult to select parameters	Multimodal or unimodal functions
DE/best/2	$v_i = x_{best}^t + F(x_{r1} - x_{r2} + x_{r3} - x_{r4})$	Similar to DE/ best /1, adding random individuals can maintain population diversity and have strong anti-rotation ability	Multimodal functions, rotation problems
DE/rand/2	$v_i = x_{r1} + F(x_{r2} - x_{r3} + x_{r4} - x_{r5})$	Similar to DE/ rand /1, the number of random individuals increases, which improves the search success rate, but the calculation time increases	Higher dimensional multimodal functions

If the traditional differential evolution algorithm is static, the purpose of the improvement is to make the algorithm dynamic and flexible, and can dynamically choose the mutation mode according to the search environment. EMBBO (Zhang et al. 2019a), TDBBO (Zhao et al. 2019) and DCGBBO (Sang et al. 2021) all add differential mutation operator to improve the search ability of original BBO. However, these algorithms only design dynamic transformation modes with two mutation modes at most. The algorithm is not universal and can not search effectively when faced with multi-modal problem. These algorithms are actually static, that is, the population cannot adaptively adjust the search direction. In order to make the population select the mutation formula intelligently according to the current evolution situation, we set up a loop with feedback mechanism. The algorithm is formed into a closed-loop evolution process so that the search direction can be adjusted according to the current population standard deviation. The specific operation is: the standard deviation of habitat suitability value is taken as feedback information, so the mutation mode is dynamically selected according to the standard deviation of the population in each iteration to form an evolution process with feedback mechanism. The standard deviation and relative standard deviation of the population are calculated as follows:

$$\sigma = \sqrt{\frac{1}{N} \cdot \sum_{i=1}^{N} (HSI_i - \mu)^2} \tag{10}$$

$$\begin{cases} \tau = \dfrac{\sigma}{\max(\sigma_i)} \\ \sigma_i = \sqrt{(HSI_i - \mu)^2} \end{cases}, i = 1, 2, \cdots, NP. \tag{11}$$

Where, μ is the mean value of fitness of all individuals in the population, and τ is a temporary variable to measure the relative size of the population standard deviation. According to Eqs. (10) and (11), a differential evolution process with feedback mechanism is designed, and this calculation step is used to replace the mutation operator in the original BBO. Algorithm 5 describes the specific calculation process of feedback differential evolution.

The standard deviation of the population can be used to feedback the degree of aggregation and dispersion of the population, and the mutation mode of the algorithm can be dynamically adjusted in real time to ensure that the population always carries out global search in the direction of the optimal solution. At the same time, the powerful search ability of differential evolution also enables BBO to make full use of the existing population information and constantly develop new individuals in the search space. This feedback mechanism greatly improves the convergence velocity and precision of the algorithm, and improves the comprehensive performance of the algorithm.

Algorithm 5 Feedback differential evolution mechanism in ZGBBO

```
for each habitat $x_i$
    if rand(0,1)<Cr
        switch $\tau$
            case $\tau$ >=0.8
                DE/best/1
            case $\tau$ >=0.6 && $\tau$ <0.8
                DE/best/2
            case $\tau$ >=0.4 && $\tau$ <0.6
                DE/rand-to-best/1
            case $\tau$ >=0.2 && $\tau$ <0.4
                DE/rand/1
            case $\tau$ <0.2
                DE/rand/2
        end switch
    end
end for
```

3.4 Greedy selection for the best individual

In order to ensure that the population always searches in a better direction, we do greedy selection of the optimal individuals in each generation in the algorithm, so that the optimal individuals in each generation of the population will not be worse than the previous generation. Algorithm 6 gives the specific operation of the greedy selection strategy for optimal individuals.

Algorithm 6 Greedy selection strategy for optimal individuals in ZGBBO

```
if fitness( $x_{best}^{t}$ ) ≤ fitness( $x_{best}^{t-1}$ )
    $x_{best}^{t} = x_{best}^{t-1}$
end if
```

The reason for designing Algorithm 6 is that the optimal individuals of the current population are used in both Eq. (8) and Algorithm 5. Only when the best individual of each generation does not degenerate can we ensure that the population does not degenerate during evolution. If the optimal individual of the previous generation is better than the optimal individual of the current generation, the population will search in the opposite direction and start to reverse evolution, which not only affects the convergence accuracy of the algorithm, but also leads to the blind search of the population in the solution space. INSBBO (An et al. 2020) uses an elite storage strategy to preserve superior individuals. But the way it works is that the elite of the previous generation directly replaces the inferior individuals of the current generation, which means that there are likely to be several identical individuals in the population, thus reducing the diversity of the population. And the elite storage strategy does not guarantee that the population will search in a better direction. Therefore, this paper uses the optimal individual greedy choice to replace the elite storage.

ZGBBO removes the mutation operator of the original BBO, reduces the computational cost, and replaces it with a differential evolution mechanism with feedback loop. We improve the migration operator of the original algorithm, design a convex migration operator, and incorporate a opposition-based learning strategy. Example learning method is adopted to select excellent individuals for migration, which reduces the calculation of habitat emigration rate and further reduces the computational complexity. In Algorithm 7, we describe in detail the pseudo-code of ZGBBO's calculation process. In each iteration of the original BBO algorithm, the immigration rate and emigration rate of each habitat in the population need to be recalculated. But the immigration rate and emigration rate are based on rankings. In other words, the immigration rate and emigration rate are only related to the ranking of habitats. According to Eq. (3), when the ranking of an individual is determined, the immigration and emigration rate of the individual can be determined. Therefore, in the ZGBBO algorithm, only the immigration rate of all habitats in the population is calculated once.

Algorithm 7 ZGBBO calculation process

```
initialization parameters, including $NP$, $S_{\max}$, $Cr$, etc
use Eqs. (1) and (2) to generate the initial population and calculate the species number $S_i$ in each habitat
use Eq. (3) to calculate the immigration rate $\lambda_i$ of each habitat
calculate the fitness value HSI for each habitat in the initial population
while (the termination condition is not met)
    rank all habitats according to their HSI from highest to lowest
    perform Algorithm 4 to complete the migration
    calculate the standard deviation and relative standard deviation of the population through Eqs. (10) and (11)
    perform Algorithm 5 and use feedback differential evolution to search for optimization
    calculate the fitness value HSI for each habitat in the current population
    perform Algorithm 6 to retain the optimal individual
end while
```

4 ZGBBO convergence proof

In this chapter, we will use two methods to prove the convergence of ZGBBO algorithm. In section 4.1, Markov model is used to prove the convergence of ZGBBO. In section 4.2, a new proof method is proposed to prove that ZGBBO has global convergence again by establishing a sequence convergence model.

4.1 Markov model

This section uses Markov model to prove the convergence of ZGBBO algorithm. The following explanations should be made: (1) the ZGBBO algorithm proposed in this paper is based on real number coding and is proposed for continuous variables, so when proving global convergence, the search space of the algorithm is a continuous state space. (2) The improved BBO is composed of selection, migration and feedback differential evolution, which is independent of the maximum number of iterations, and the population size of the algorithm is fixed. Therefore, it can be considered that the optimization process of ZGBBO satisfies the finite homogeneous Markov model.

Theorem 1 Stability theorem of reducible random matrix (Kingman 1981).

The n-order matrix Q is a reducible matrix, if Q can be obtained Eq. (12) by the same row and column transformations:

$$Q = \begin{bmatrix} C & O \\ R & T \end{bmatrix}. \tag{12}$$

Where, C is a primitive matrix of order $m(m \le n)$, while R and T are non-zero matrices of order $n-m$. If

$$Q^{\infty} = \lim_{k\to\infty} Q^k = \lim_{k\to\infty} \begin{bmatrix} C^k & O \\ \sum_{i=1}^{k-1} T^i R C^{k-i} & T^k \end{bmatrix} = \begin{bmatrix} C^{\infty} & O \\ R^{\infty} & O \end{bmatrix}, \tag{13}$$

then the matrix Q^{∞} is a stable random matrix, and $Q^{\infty}{=}1'Q^{\infty}$, $Q^{\infty} = Q^0 \cdot Q^{\infty}$ is uniquely determined and independent of the initial distribution (Kingman 1981).

If Q^{∞} is a stable random matrix, then Q^{∞} satisfies the following conditions:

$$Q^{\infty} = (q_{ij})_{n\times n} = \begin{cases} q_{ij} > 0, 1 \le i \le n, 1 \le j \le m \\ q_{ij} = 0, 1 \le i \le n, m \le j \le n \end{cases}. \tag{14}$$

The population of ZGBBO is randomly divided into w subsets, so when the iteration number is t, the population can be expressed as $X(t) = \{x_1(t), x_2(t), \cdots, x_w(t)\}$. According to the above description, for sub-population $x_i(t), i = 1,2,\cdots,w$, it can be equivalent to a state on finite Markov chain, and $p_i(t)$ represents the probability of being in $x_i(t)$. Then the probability of moving from state $x_i(t)$ to state $x_j(t)$ can be represented by p_{ij}:

$$\begin{cases} p_{ij} = 0, & if \;\; i < j \\ p_{ij} > 0, & else \end{cases}. \tag{15}$$

Therefore, the state transition matrix P of Markov chain based on ZGBBO algorithm is a lower triangular matrix:

$$P=\begin{bmatrix} p_{11} & 0 & \cdots & 0 \\ p_{21} & p_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ p_{w1} & p_{w2} & \cdots & p_{ww} \end{bmatrix}. \tag{16}$$

We make:

$$S_1=\begin{bmatrix} p_{22} & \cdots & 0 \\ & \ddots & \\ p_{w2} & \cdots & p_{ww} \end{bmatrix}, \tag{17}$$

$$S_2=(p_{21},p_{31},\cdots,p_{w1})^{\mathrm{T}}. \tag{18}$$

Then, we can get:

$$P=\begin{bmatrix} p_{11} & O \\ S_2 & S_1 \end{bmatrix}. \tag{19}$$

According to Eqs. (15) to (18), S_1,S_2 are non-zero matrices. The sum of probabilities of each line of Markov transition matrix is 1, so $p_{11}=1$, that is, p_{11} is the first-order primitive matrix. Therefore, Markov state transition matrix P meets the condition requirements of Theorem 1, so P is a reducible random matrix.

Therefore, the following formula can be obtained:

$$\begin{aligned} P^{\infty} &= \lim_{t\to\infty}\begin{bmatrix} p_{11} & O \\ S_2 & S_1 \end{bmatrix}^{t} \\ &= \lim_{t\to\infty}\begin{bmatrix} p_{11}{}^{t} & o \\ \sum_{i=1}^{t-1} S_1^{i} S_2 p_{11}^{t-i} & S_1^{t} \end{bmatrix} \\ &= \begin{bmatrix} p_{11}^{\infty} & o \\ S_2^{\infty} & O \end{bmatrix}. \end{aligned} \tag{20}$$

We know that $p_{11}=1$, so we have $p_{11}^{\infty}=p_{11}=1$, and $S_1^{\infty}=O$. The sum of probabilities of each line of Markov state transition matrix is 1, so there must be $S_2^{\infty}=(1,1,\cdots,1)^{\mathrm{T}}$. Then the following formula can be obtained:

$$P=\begin{bmatrix} 1 & 0 & \cdots & 0 \\ 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & \cdots & 0 \end{bmatrix}. \tag{21}$$

According to Eq. (21), when the evolutionary algebra $t\to\infty$, the probability $p_{i1}^{\infty}=1$, where, $i=1,2,\cdots,w$. Therefore, regardless of the initial state, after countless iterations, each state $x_i(t)$ will converge to the global optimal solution with probability 1. That is, the following formula holds:

$$\lim_{t\to\infty} P(\mathit{fitness}(t)=f(*))=1. \tag{22}$$

Where, $\mathit{fitness}(t)$ is the optimal fitness value of sub-population $x_i(t)$, $f(*)$ is the global optimal value, and $P(\bullet)$ is the probability that the optimal value of the t-th iteration of the algorithm converges to the global optimal value. According to Eq. (22), the ZGBBO algorithm proposed in this paper must converge to the global optimal value after several iterations, so the algorithm has global convergence.

4.2 Sequence convergence model

In this section, we will prove the global convergence of ZGBBO with a new method. For a global optimization problem, assuming that its optimal solution is x^{*}, then $f(x^{*})$ is the global optimal value. The optimal solution of ZGBBO algorithm 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 ZGBBO algorithm can find the global optimal value $f(x^{*})$ is that a certain $f(x_{best}^{t})$ is in the δ domain of the global optimal value $f(x^{*})$, that is, $\| f(x_{best}^{t})-f(x^{*})\| \le \delta$.

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

$$X_{best}=\{x_{best}^{1},x_{best}^{2},\cdots,x_{best}^{t},\cdots,x_{best}^{T}\}. \tag{23}$$

Where, T represents the maximum number of iterations. Thus, sequence A can be constructed according to Eq. (24) :

$$\begin{cases} A = \{a_1, a_2, \cdots, a_T\} \\ a_i = f(x^i_{best}), i = 1, 2, \cdots, T \end{cases}. \tag{24}$$

As can be seen from section 3.4, since greedy selection is adopted for the optimal individual of each generation in ZGBBO, the optimal value of each generation population must be better or equivalent that of the previous generation. Therefore, the following formula must be true:

$$a_1 \le a_2 \le \cdots \le a_T. \tag{25}$$

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. (26) is used to express the probability that the optimal value $f(x^t_{best})$ of the current population converges to the global optimal value:

$$p_t = \{\| f(x^t_{best}) - f(x^*) \| \le \delta\}, t = 1, 2, \cdots, T. \tag{26}$$

According to Eq. (25), the following relationship must exist:

$$p_1 \le p_2 \le \ldots \le p_t, t = 1, 2, \cdots, T. \tag{27}$$

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

$$P_t = (1 - p_1)(1 - p_2) \cdots (1 - p_t). \tag{28}$$

From Eq. (27), it can be seen that p_t is monotone and does not decrease, so the following formula is true:

$$\begin{aligned} P_t &= (1 - p_1)(1 - p_2) \cdots (1 - p_{t-1})(1 - p_t) \\ &\le (1 - p_1)(1 - p_1) \cdots (1 - p_1)(1 - p_1) \\ &= (1 - p_1)^t \end{aligned} \tag{29}$$

And since p_1 is the probability, so $0 \le p_1 \le 1$, then we have $0 \le (1\text{-}p_1) \le 1$. After many iterations, Eq. (30) must hold.

$$\lim_{t \to \infty} (1 - p_t)^t = 0 \tag{30}$$

According to Eq. (30), after a large number of iterations, the probability that the algorithm does not converge to the optimal value is 0. Therefore, as the number of iteration t increases, ZGBBO will eventually converge to the global optimal value $f(x^*)$ in the form of probability 1, so the proof is completed.

5 Experiment and analysis

In order to test and verify the practicality and advancement of the ZGBBO algorithm proposed in this paper, a series of comparative experiments are carried out in this chapter. Firstly, the parameter sensitivity of the proposed algorithm is analyzed. Secondly, the ZGBBO is compared with its own variant algorithm, that is, the new algorithm obtained by ZGBBO without any improved strategy is compared and analyzed with the complete ZGBBO, in order to prove the effectiveness and necessity of the three improved strategies proposed in this paper. Then, ZGBBO is compared with the original BBO and other 6 excellent BBO variants proposed in the resent 5 years. Finally, ZGBBO is compared with 6 new state-of-the-art population intelligence evolutionary algorithms proposed in the resent 3 years to fully prove its advanced nature. All experiments in this paper were carried out on the computer with Intel(R) Core(TM) I5-8500 CPU @ 3.00ghz processor, 8.00 GB running memory and 64-bit operating system, and the development environment was Matlab R2020a.

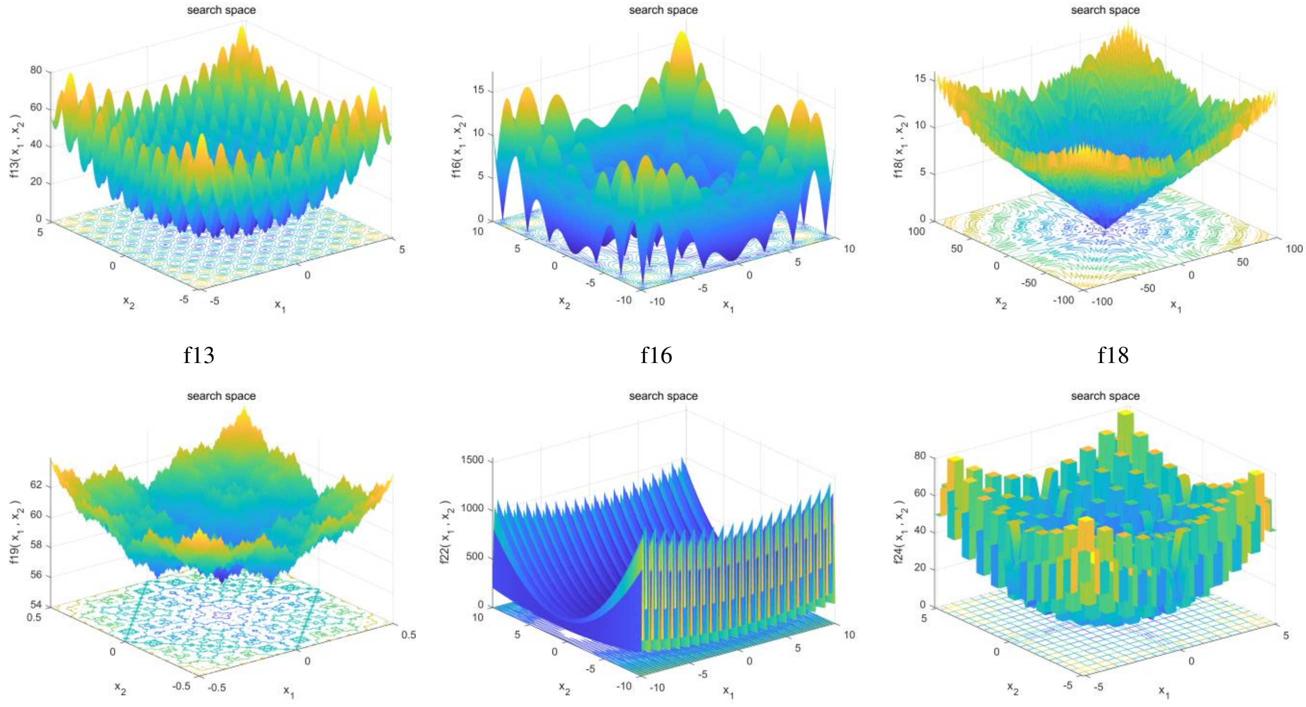
At present, the test functions of evolutionary algorithms are mainly derived from Congress on Evolutionary Computation (CEC), the most widely used are CEC2013 (Liao et al. 2013), CEC2014 (Erlich et al. 2014) and CEC2017 (Awad et al. 2017). This paper summarizes and selects 24 different types of high-dimensional benchmark functions to test, and the specific information is shown in Table 3.

Table 3 Benchmark Functions

F	Name	Function	Search Space	f(x*)
f1	SumSquares	$f(x) = \sum_{i=1}^{D} i x_i^2$	$[-10,10]^D$	0
f2	Schwefel 2.22	$f(x) = \sum_{i=1}^{D} \lvert x_i \rvert + \prod_{i=1}^{D} \lvert x_i \rvert$	$[-10,10]^D$	0
f3	Zakharov	$f(x) = \sum_{i=1}^{D} x_i^2 + (\sum_{i=1}^{D} 0.5 i x_i)^2 + (\sum_{i=1}^{D} 0.5 i x_i)^4$	$[-5,10]^D$	0

f4	Schwefel 2.21	$f(x)=\max_{i=1}^{D}\{\lvert x_i\rvert\}$	$[-100,100]^D$	0
f5	Schwefel 1.2 with noise	$f(x)=\sum_{i=1}^{D}(\sum_{j=1}^{i}x_j)^2\times(1+0.4\lvert N(0,1)\rvert)$	$[-100,100]^D$	0
f6	Schwefel 1.2	$f(x)=\sum_{i=1}^{D}(\sum_{j=1}^{i}x_j)^2$	$[-100,100]^D$	0
f7	Quartic	$f(x)=\sum_{i=1}^{D}ix_i^4+random[0,10)$	$[-100,100]^D$	0
f8	Dixon-Price	$f(x)=(x_1-2)^2+\sum_{i=2}^{D}i(2x_i^2-x_i-1)^2$	$[-10,10]^D$	0
f9	SumPower	$f(x)=\sum_{i=1}^{D}\lvert x_i\rvert^{i+1}$	$[-1,1]^D$	0
f10	Exponential	$f(x)=\exp(0.5\sum_{i=1}^{D}\lvert x_i\rvert)-1$	$[-1.28,1.28]^D$	0
f11	Griwank	$f(x)=\frac{1}{4000}[\sum_{i=1}^{D}(x_i-100)^2]-[\prod_{i=1}^{D}\cos(\frac{x_i-100}{\sqrt{i}})]+1$	$[-600,600]^D$	0
f12	Step	$f(x)=\sum_{i=1}^{D}\lfloor x_i+0.5\rfloor^2$	$[-100,100]^D$	0
f13	Rastrigin	$f(x)=\sum_{i=1}^{D}[x_i^2-10\cos(2\pi x_i)+10]$	$[-5.12,5.12]^D$	0
f14	Noncontinuous Rastrigin	$f(x)=\sum_{i=1}^{D}[z_i^2-10\cos(2\pi z_i)+10],\ z_i=\begin{cases}x_i, & \lvert x_i\rvert<0.5\\ \text{round}(2x_i)/2, \text{else}\end{cases}$	$[-5.12,5.12]^D$	0
f15	Ackley	$f(x)=-20\exp(-0.2\sqrt{\sum_{i=1}^{D}\frac{x_i^2}{D}})-\exp[\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi x_i)]+20+e$	$[-32,32]^D$	0
f16	Alpine	$f(x)=\sum_{i=1}^{D}\lvert x_i\sin(x_i)+0.1x_i\rvert$	$[-10,10]^D$	0
f17	Elliptic	$f(x)=\sum_{i=1}^{D}(10^6)^{\frac{i-1}{D-1}}x_i^2$	$[-100,100]^D$	0
f18	Salomon	$f(x)=-\cos(2\pi\sqrt{\sum_{i}^{D}x_i^2})+0.1\times\sqrt{\sum_{i}^{D}x_i^2}+1$	$[-100,100]^D$	0
f19	Weierstrass	$f(x)=\sum_{i=1}^{D}\{\sum_{k=0}^{k_{\max}}[a^k\cos(2\pi b^k(x_i+0.5))]\}-D\sum_{k=0}^{k_{\max}}[a^k\cos(2\pi b^k\times0.5)], a=0.5, b=3, k_{\max}=20$	$[-0.5,0.5]^D$	0
f20	Penalized 1	$f(x)=0.1\{\sin^2(3\pi x_1)+\sum_{i=1}^{D}(x_i-1)^2[1+\sin^2(3\pi x_i+1)]+(x_D-1)^2[1+\sin^2(2\pi x_D)]\}+\sum_{i=1}^{D}u(x_i,5,100,4)$ $u(x_i,a,k,m)=\begin{cases}k(x_i-a)^m, x_i>a\\0,=-a\le x_i\le a\\k(-x_i-a)^m, x_i<-a\end{cases}$	$[-50,50]^D$	0
f21	Penalized 2	$f(x)=0.1\{\sin^2(3\pi x_1)+\sum_{i=1}^{D-1}(x_i-1)^2[1+\sin^2(3\pi x_{i+1})]+(x_D-1)[1+\sin^2(2\pi x_D)]\}+\sum_{i=1}^{D}u(x_i,5,100,4)$ $u(x_i,a,k,m)=\begin{cases}k(x_i-a)^m, x_i>a\\0,=-a\le x_i\le a\\k(-x_i-a)^m, x_i<-a\end{cases}$	$[-50,50]^D$	0
f22	Levy	$f(x)=\sum_{i=1}^{D-1}(x_i-1)^2[1+10\sin^2(3\pi x_{i+1})]+\sin^2(3\pi x_1)+\lvert x_D-1\rvert^2[1+\sin^2(3\pi x_D)]$	$[-100,100]^D$	0
f23	Bohachevsky_2	$f(x)=\sum_{i=1}^{D-1}[x_i^2+2x_{i+1}^2-0.3\cos(3\pi x_i)\cos(3\pi x_{i+1})+0.3]$	$[-100,100]^D$	0
f24	NCRastrigin	$f(x)=\sum_{i=1}^{D}[y_i^2-10\cos(2\pi y_i)+10],\ y_i=\begin{cases}x_i,\lvert x_i\rvert<0.5\\ round(2x_i)/2,\lvert x_i\rvert\ge0.5\end{cases}$	$[-5.12,5.12]^D$	0

Among them, f1-f10 are static single objective functions with only one optimal value, which is usually used to test the convergence speed and optimization accuracy of the algorithm, and the execution ability of the algorithm. f11-f24 are high-dimensional multi-modal functions, and the number of their local minimum increases exponentially with the dimension. It is mainly used to test the ability of the algorithm to jump out of the local optimal state and the global search ability of the algorithm. In order to fully prove the algorithm testing ability of these multi-modal functions selected in this paper, we take the two-dimensional search space as an example to plot figures of several representative multi-modal functions, as shown in Fig. 4.



f13 f16 f18

f19 f22 f24

Fig. 4 The two-dimensional graphs of multi-modal functions

It can be seen from the six sub-graphs of Fig. 4 that these multi-modal functions have many local minima in the two-dimensional problem environment. As the dimension of the problem increases, the number of these local optimal values will increase exponentially. For example, the function f22 has only one minimum, but it is very shallow at the minimum and there are many local minima nearby, so it is difficult to find the theoretical minimum with high accuracy. Searching for the optimal solution on these irregular multi-modal problems is a huge challenge to the algorithms. Therefore, these benchmark functions can be used to effectively test the optimization performance of the algorithms.

5.1 Parameter sensitivity analysis

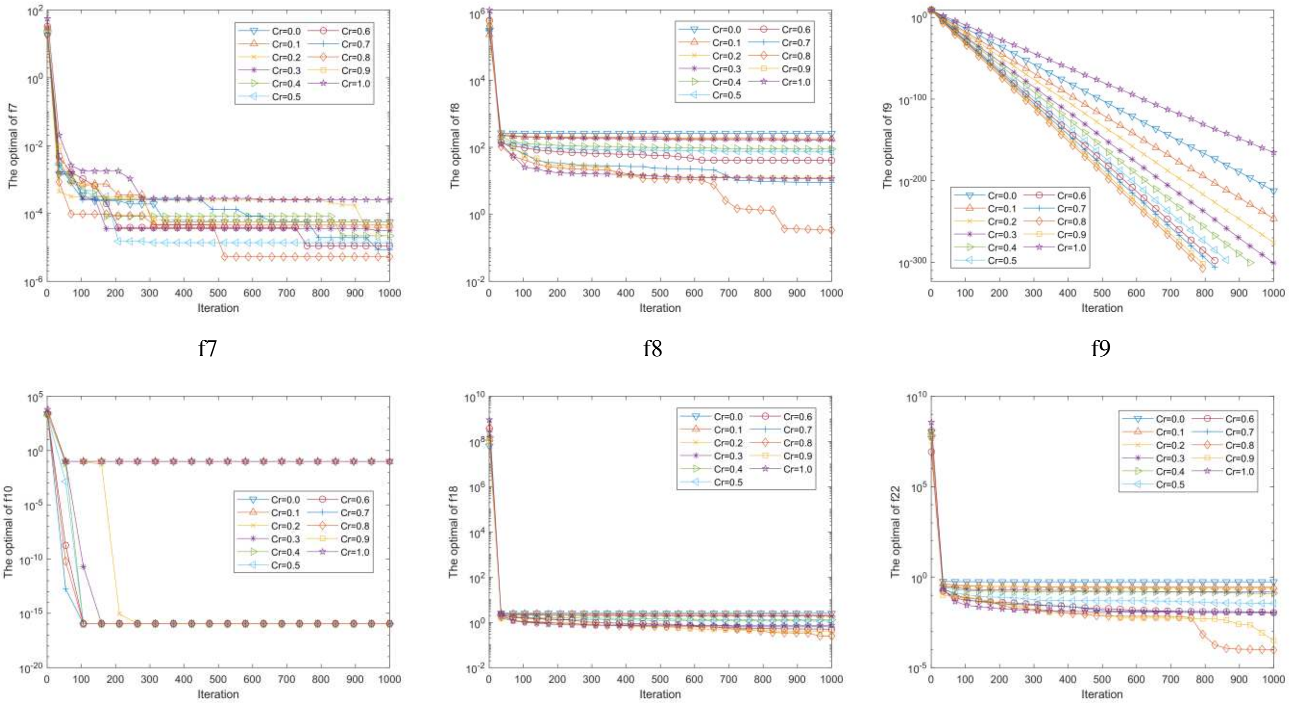
In this section, we will analyze the parameter sensitivity of the proposed algorithm. An appropriate parameter value is very important to the optimization performance of the algorithm, and even affects the ability of the algorithm to solve most problems. According to Algorithm 5, *Cr* is a parameter that needs to be initialized in addition to the original parameter. *Cr* means the individual crossover rate in feedback differential evolution mechanism, which can also be understood as mutation rate in our algorithm. For convenience, the maximum immigration rate *I* and emigration rate *E* are set as 1, and the optimal value of *Cr* is determined by adjusting the value of *Cr*. *Cr* value is adjusted and is set to 0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, then the optimal values of the 24 benchmark functions in Table 3 are searched in a 50-dimensional problem environment. Other parameters for this experiment are set as follows: the population size NP=5D (D is the problem dimension), the maximum species number $S_{\max}$=2NP, the maximum iteration times T=1000. In order to briefly illustrate the experimental results, the specific experimental data table is not listed and only the Friedman test results with different *Cr* values are shown. The Friedman test results are shown in Table 4.

Table 4 Friedman test results of different *Cr* values

Cr	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	**0.8**	0.9	1.0
Mean Rank	4.5190	4.2315	4.1983	4.1444	4.0155	3.8575	3.8115	2.7838	**2.5714**	3.3435	4.3680
Final Rank	11	9	8	7	6	5	4	2	**1**	3	10

From Table 4, it can be found that when *Cr* value is 0.8, its Friedman test result is the best and its average ranking is the highest

(2.5714). When *Cr* value is 0.0 or 1.0, the experimental results are the worst. It shows that the lack of feedback differential evolution mechanism or too frequent use it will seriously reduce the optimization efficiency of the algorithm. In addition, the value of *Cr* also has a very serious influence on the optimization results of some problems. As shown in Fig. 5, for example, on the functions f7, f8, f9 f10, f18 and f22, different values of *Cr* will significantly affect the optimization performance of the algorithm. On these functions, *Cr* equals 0.8 is the best choice. So the parameter of our algorithm is suggested that *Cr* is 0.8.



f7 f8 f9

f10 f18 f22

Fig. 5 Optimization results of functions f7, f8, f9, f10, f18 and f22 on different crossover rates *Cr*

5.2 Comparison between ZGBBO and its own variants

In this paper, ZGBBO algorithm is compared with its own three variants to prove the necessity of three main improvement strategies. Since the greedy selection of optimal individual in section 3.4 makes the optimal value of each generation population not worse than that of the previous generation, which improves the convergence accuracy of the algorithm, the improved strategy in Section 3.4 need not be verified. The three variant algorithms used for comparison in this section are the algorithm formed by the lack of one of the three improved mechanisms in section 3.1, 3.2 and 3.3. The specific information is shown in Table 5. There are three variants: ZGBBO_1, which lacks the example learning method in section 3.1, that is, BBO algorithm using roulette selection operator, improved migration operator and feedback difference mechanism; ZGBBO_2, which lacks the improved migration operator in section 3.2, that is, BBO algorithm using example learning method, original migration operator and feedback difference mechanism; ZGBBO_3, lacks the feedback difference mechanism in section 3.3, that is, BBO algorithm using example learning method, improved migration operator and random mutation operator.

Table 5 The composition of three ZGBBO's variant algorithms

Name	Algorithm constitute	Missing strategy
ZGBBO_1	BBO+ section 3.2+ section 3.3	Example learning method
ZGBBO_2	BBO+ section 3.1+ section 3.3	Convex migration mechanism and opposition-based learning
ZGBBO_3	BBO+ section 3.1+ section 3.2	Feedback differential evolution

5.2.1 Optimize performance comparison

The maximum immigration rate *I* and maximum emigration rate *E* of the four ZGBBO algorithms are 1, the population size *NP*=5*D* (*D* is the problem dimension), the maximum species number $S_{\max}$=2*NP*, the maximum iteration times *T*=1000, and the maximum mutation rate $m_{\max} = 0.05$ of ZGBBO_3. In order to avoid contingency and keep the experiment scientific, four algorithms will independently run for 50 times on each test function (*D*=50). The mean value (Mean) and standard deviation (Std) of the results of 50 times will be counted. At the significance level of $\alpha = 0.05$, Wilcoxon rank sum test is performed (Joaquín Derrac et al. 2011). The comparison results of the four algorithms are shown in Table 6. The bold data represents the optimal results of the four algorithms. The Wilcoxon rank sum test results of the 24 test functions are (w/t/l), and the representative meaning is w(+:win)/t(=:tie)/l(-:lose). For each test function in the table, "-" means that the performance of the contestant algorithm is worse than that of ZGBBO, "+" means that the performance of the contestant algorithm is superior, and "=" means that the result of the contestant algorithm is the same as that of ZGBBO, that is, there is no statistical significance in their performance difference.

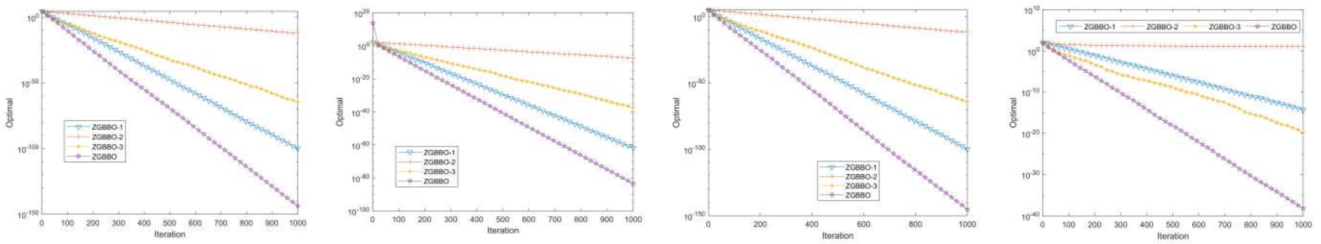
Table 6 Comparison between ZGBBO and three self-variant algorithms

F	ZGBBO_1		ZGBBO_2		ZGBBO_3		ZGBBO	
	Mean	**Std**	**Mean**	**Std**	**Mean**	**Std**	**Mean**	**Std**
f1	3.54E-141	2.01E-281	2.01E-19	6.69E-39	4.59E-04	7.94E-08	**1.70E-146**	**9.27E-293**
f2	6.80E-81	5.41E-161	8.90E-15	5.94E-29	5.23E-03	1.95E-06	**1.09E-83**	**7.62E-167**
f3	7.84E-141	2.10E-280	2.87E-19	3.74E-38	7.69E-04	4.08E-08	**1.84E-146**	**2.39E-292**
f4	2.72E-37	3.13E-74	3.73E+01	5.56E+01	1.01E+00	2.14E-02	**9.47E-38**	**2.84E-75**
f5	2.90E-264	0.00E+00	1.14E-28	1.11E-57	2.05E+00	1.45E+00	**1.16E-272**	**0.00E+00**
f6	3.07E-137	4.37E-273	6.92E-17	7.70E-34	2.96E-01	5.54E-03	**9.03E-142**	**1.93E-282**
f7	1.26E-03	2.36E-07	3.55E-02	6.34E-05	7.34E-03	2.93E-06	**1.19E-03**	**1.52E-07**
f8	2.09E-08	3.32E-16	8.45E+00	2.36E+01	4.50E-03	1.93E-05	**2.28E-11**	**2.54E-22**
f9	**0.00E+00**	**0.00E+00**	1.85E-07	1.03E-13	3.90E-13	6.09E-25	**0.00E+00**	**0.00E+00**
f10	1.78E-16	9.86E-32	6.37E-15	7.02E-30	3.57E-04	8.49E-09	**1.48E-16**	**1.64E-33**
f11	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	6.21E-03	2.02E-05	**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**
f13	**0.00E+00**	**0.00E+00**	3.54E+01	3.93E+02	9.68E-04	3.90E-07	**0.00E+00**	**0.00E+00**
f14	**0.00E+00**	**0.00E+00**	1.85E+01	4.51E+01	9.11E-04	2.34E-07	**0.00E+00**	**0.00E+00**
f15	**4.44E-15**	**0.00E+00**	1.20E+01	1.24E+01	8.35E-03	7.43E-06	**4.44E-15**	**0.00E+00**
f16	3.77E-16	7.12E-31	3.08E+00	3.23E+00	9.41E-04	1.18E-07	**3.81E-62**	**4.34E-123**
f17	2.50E-137	1.28E-273	2.70E-15	2.24E-30	3.40E+01	1.58E+03	**3.92E-142**	**5.64E-284**
f18	1.67E-01	3.33E-03	7.00E-01	1.00E-02	1.26E+00	1.80E-02	**1.20E-01**	**2.00E-03**
f19	**0.00E+00**	**0.00E+00**	2.64E-02	2.25E-04	1.31E-01	7.17E-04	**0.00E+00**	**0.00E+00**
f20	2.26E-11	1.46E-21	3.78E+00	6.66E+01	3.55E-06	5.90E-12	**4.05E-15**	**1.47E-29**
f21	8.79E-03	3.86E-04	2.21E-03	2.41E-05	4.03E-05	7.78E-10	**1.47E-09**	**2.44E-18**
f22	7.39E+00	7.76E+00	5.34E+04	1.21E+08	1.82E+00	2.02E+00	**1.74E-01**	**3.16E-02**
f23	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	2.38E-01	1.17E-01	**0.00E+00**	**0.00E+00**
f24	**0.00E+00**	**0.00E+00**	2.07E+01	2.83E+01	1.18E-03	2.96E-07	**0.00E+00**	**0.00E+00**
b/w/l	0/9/15		0/3/21		0/1/23			

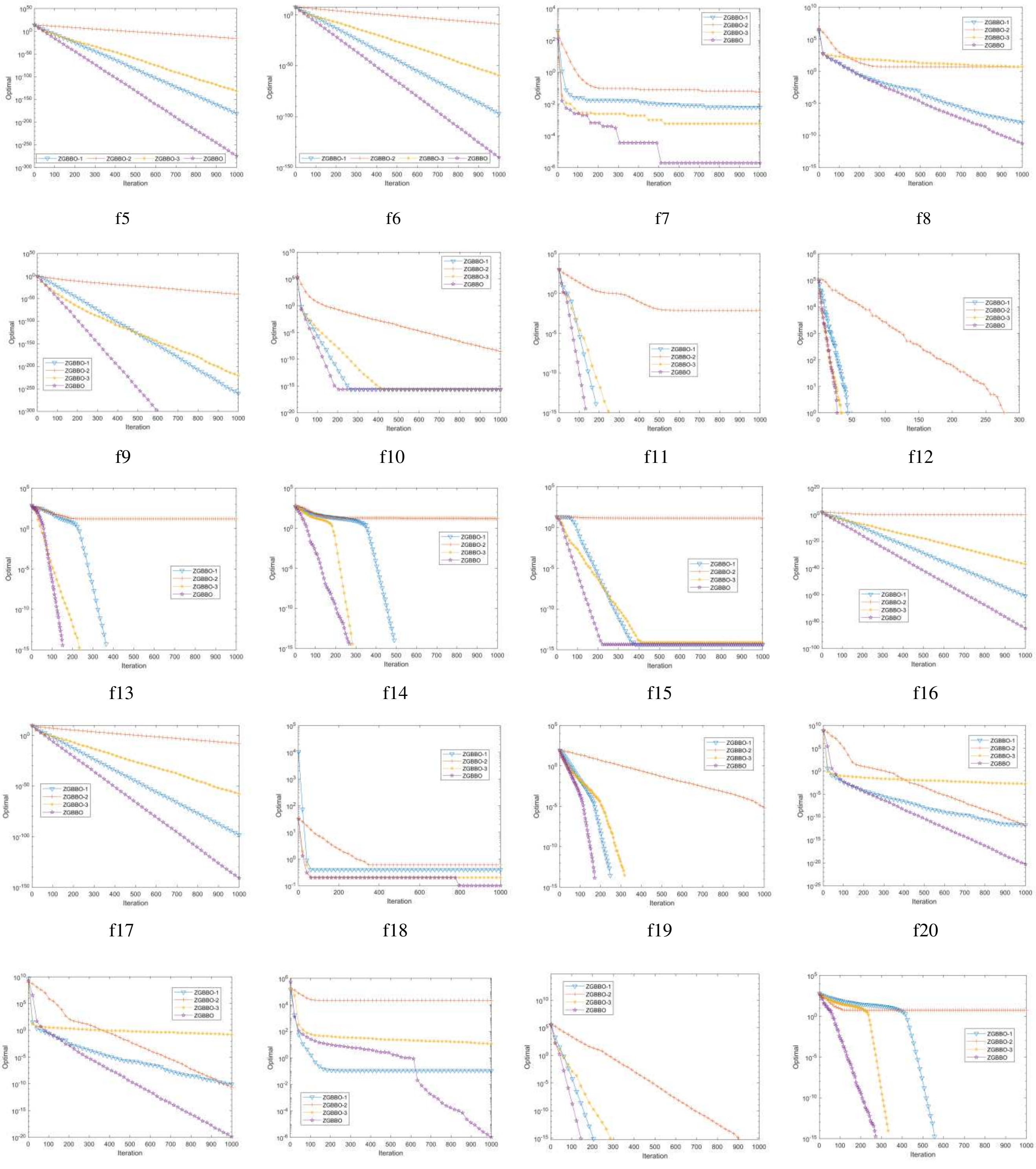
As can be seen from Table 6, the overall optimization performance of the three ZGBBO variants is not as good as that of ZGBBO. ZGBBO_1 without the example learning method has the same results on 9 problems as ZGBBO, but is inferior to ZGBBO on 15 problems. ZGBBO_2, which lacks convex migration mechanism and opposition-based learning, tied with ZGBBO on 3 problems, and is inferior to ZGBBO on 21 problems. The convergence of ZGBBO_3 without feedback differential evolution is the same as that of ZGBBO in only one problem, and the convergence accuracy is not as high as that of ZGBBO on the remaining 23 problems. None of the three variants showed better performance than ZGBBO, indicating that the lack of any improvement strategy would reduce the search capability of the algorithm. ZGBBO, which integrates all the improved mechanisms, shows obvious advantages on 24 test functions and has higher convergence accuracy, which indicates that the improved strategies proposed in this paper are effective and can improve the performance of the algorithm. Moreover, all three improvement mechanisms are indispensable.

5.2.2 Comparison of convergence rates

This section compares the convergence speed of ZGBBO and the three variants on different test functions, so as to compare the algorithm performance. The specific operation is to take the optimal value of each algorithm and the search results of each generation in the 50 times of operation in section 5.2.1, and plot the optimal convergence images of the four algorithms on different test functions (*D*=50), as shown in Fig. 6.



f1 f2 f3 f4



f5 f6 f7 f8

f9 f10 f11 f12

f13 f14 f15 f16

f17 f18 f19 f20

f21 f22 f23 f24

Fig. 6 Convergence of ZGBBO and three self-variant algorithms in different test functions

As can be seen from Fig. 6, ZGBBO's convergence speed and convergence accuracy are significantly higher than those of the other three self-variant algorithms, no matter on single-modal functions or multi-modal functions. Therefore, the three improved strategies are all resultful. The overall convergence situation of the four variants on different test functions is roughly the same, ZGBBO has the fastest convergence speed and the highest accuracy, while ZGBBO_2 has the slowest convergence speed. From Fig. 6, the convergence of the four comparison algorithms is basically the same on the single-peak functions f1-f9 and multi-modal functions f16, f17 and f21. The convergence accuracy of ZGBBO is rapidly improved and 20 to 100 exponential levels higher than that of the other three variants. The optimization performance of ZGBBO_1 and ZGBBO_3 is between ZGBBO and ZGBBO_2. For the multi-modal functions f9-f11, f13-f22, f24, ZGBBO_2 can not jump out of the local optimal solution, and the convergence accuracy of it is low. Therefore, the convex migration mechanism and opposition-based learning strategy adopted in this paper can not only accelerate the convergence speed of the algorithm, but also effectively help the algorithm to jump out of the local optimal state. In the case of single peak, the convergence speed of ZGBBO_3 without feedback difference mechanism is obviously faster than that of ZGBBO_1 without example learning method. However, in the case of multi-modal problem, ZGBBO_3 without feedback differential mechanism can not select the mutation mode intelligently according to the population information, so it is difficult to jump out of the local optimal solution and the search speed is slower than that of ZGBBO_1. It can be seen that the feedback difference mechanism can help the intelligent evolution of the population, change the direction of optimization and improve the calculation accuracy. In addition, ZGBBO_1 without example learning method has faster convergence speed than ZGBBO_2 on all test functions and better search performance than ZGBBO_3 on functions f1-f3, f5-f11 and f16-f23. However, the convergence speed and accuracy of ZGBBO_1 are always worse than that of ZGBBO. Therefore, the example learning method effectively improves the convergence speed of the algorithm and enables the algorithm to converge to the global optimal value more quickly.

In general, ZGBBO is the best algorithm among the four algorithms, and its competitiveness is significantly stronger than the other three variants. Therefore, this paper is essential to the three improvement mechanisms of BBO. In addition, ZGBBO_1 algorithm and ZGBBO_3 algorithm are obviously superior to ZGBBO_2 algorithm, and show obvious advantages in the convergence accuracy of most test functions, which indicate that the convex migration operator and opposition-based learning strategy adopted in this paper have the greatest influence on the performance of the improved algorithm.

5.3 Comparison between ZGBBO and congeneric algorithms

This section compares ZGBBO with BBO and other six state-of-the-art improved BBO algorithms. Six algorithms with the strongest competitiveness and obvious improvement effect in the past five years are selected for simulation experiments to compare the overall performance. Table 7 shows the detailed information of six BBO congeneric algorithms.

Table 7 Details of six improved BBO algorithms

Article	Time	Strategy	Name
(Feng et al. 2017)	2017	BBO+ Random ring topology mixed migrati + Adaptive Powell method	PRBBO
(Xiong et al. 2018)	2018	BBO+ Brain Storm Optimization (BSO)+ Greedy select	BBOSB
(Zhao et al. 2019)	2019	BBO+ Two-stage mechanism + Gaussian mutation	TDBBO
(Zhang et al. 2019b)	2019	BBO+ Random scale differential mutation + Dynamic heuristic crossover + Elitist strategy	WRBBO
(Zhao et al. 2020)	2020	BBO+Rotation invariant migration +CMA-ES algorithm	HBBO-CMA
(Sang et al. 2021)	2021	BBO+ Dynamic cross migration + dynamic Gaussian mutation + Trigger ablation rule hierarchical tissue P-system	DCGBBO

5.3.1 Parameter settings

Consistent with section 5.2, the maximum immigration rate *I* and maximum emigration rate *E* of the eight BBO algorithms are all 1, the population size *NP*=5*D* (*D* is the problem dimension), and the maximum species number $S_{\max}$=2*NP*. Other parameters are set according to the original references for each algorithm, as shown in Table 8. As can be seen from Table 8, the ZGBBO proposed in this paper has few parameter settings, so the algorithm has good robustness. The algorithm performance is basically not affected by parameter settings.

Table 8 Parameter setting of eight BBO algorithms

Algorithm	Parameter setting
BBO	$m_{\max} = 0.05$
PRBBO	$m_{\max} = 0.005,\ \varepsilon = 0.1,\ \mathtt{limit} = 100$
BBOSB	$m_{\max} = 0.05,\ perc_e = 20,\ p_{rep} = 0.2,\ p_e = 0.6,\ p_{one} = 0.8,\ K = 25$
TDBBO	$m_{\max} = 0.01,\ \mu=0, \delta=1, c = 0.3,$
WRBBO	$\alpha_d = rand^{\beta},\ \beta = 4\sqrt{t/T},\ \alpha_h = 2\cdot(1-t/T)$
HBBO-CMA	$m_{\max} = 0.01,\ \eta = 3\ln D + 80$
DCGBBO	$m_{\max} = 0.01,\ K = 10,\ a = 0.02$
ZGBBO	$F = unifrnd(0.4, 0.9), Cr = 0.8$

5.3.2 Optimize performance comparison

This section compares the optimization performance of ZGBBO with other seven BBO algorithms. Eight algorithms search for optimal solutions on 24 test functions in Table 3. The specific operation is to set the maximum evaluation times (FEs) of each test function, and record the optimal value searched when FEs is reached. According to the setting method of FEs of test function in CEC2017, FEs of each test function is equal to 10^4D (D is the problem dimension). Each algorithm searches for the optimal values of 24 test functions in the search space of D=10, D=30 and D=50 respectively. In order to avoid contingency, each algorithm will independently run for 50 times in different problem dimensions, and the algorithm performance of the algorithm will be showed by taking the best value (Best), mean value (Mean) and standard deviation (Std) as evaluation indexes. For the three evaluation indexes, the best value reflects the convergence accuracy of the algorithm, the mean value reflects the optimization ability of the algorithm, and the standard deviation represents the stability of the algorithm. Therefore, mean value are the focus of comparison. Tables 9-11 show the optimization results of eight algorithms on 24 test functions in 10, 30 and 50 dimensions respectively, where the best results of the eight algorithms are shown in bold.

Table 9 The results of eight BBO algorithms (10-demensional benchmark functions)

F	Index	BBO	PRBBO	BBOSB	TDBBO	WRBBO	HBBO-CMA	DCGBBO	ZGBBO
	best	2.66E-03	9.97E-140	5.86E-08	1.7E-120	1.00E-136	2.59E-82	0.00E+00	0.00E+00
f1	mean	9.18E-03	1.58E-135	2.44E-06	2.80E-05	3.50E-133	8.11E-79	**0.00E+00**	**0.00E+00**
	std	3.07E-05	2.62E-269	7.08E-12	1.60E-08	5.73E-264	7.82E-156	0.00E+00	0.00E+00
	best	4.00E-02	4.42E-81	3.42E-04	4.42E-63	5.73E-77	9.35E-49	0.00E+00	4.08E-222
f2	mean	1.01E-01	4.32E-79	1.30E-03	5.56E-05	2.44E-75	4.91E-47	**0.00E+00**	4.49E-219
	std	1.02E-03	1.44E-156	5.18E-07	1.55E-07	1.30E-149	1.62E-92	0.00E+00	0.00E+00
	best	2.64E-02	6.79E-150	4.80E-09	5.3E-130	2.96E-149	9.44E-93	0.00E+00	0.00E+00
f3	mean	1.39E-01	1.28E-146	3.46E-07	6.34E-06	2.08E-145	1.73E-87	3.68E-28	**0.00E+00**
	std	1.14E-02	1.30E-291	1.52E-13	1.99E-09	3.00E-289	1.03E-172	6.78E-54	0.00E+00
	best	3.56E-01	4.48E-39	2.24E-03	1.05E-02	5.14E-42	1.04E-24	1.74E-06	2.91E-142
f4	mean	9.39E-01	2.13E-37	5.39E-03	1.02E+00	2.91E-32	3.04E-22	7.52E+00	**8.72E-137**
	std	9.41E-02	1.35E-73	4.90E-06	1.02E+00	2.07E-62	1.43E-42	2.13E+02	1.06E-271
	best	1.49E-01	7.29E-255	2.34E-12	3.2E-226	4.70E-259	1.49E-151	0.00E+00	0.00E+00
f5	mean	1.36E+01	8.40E-248	1.78E-09	2.57E-15	5.81E-251	4.49E-139	**0.00E+00**	**0.00E+00**
	std	2.74E+02	0.00E+00	1.61E-17	3.30E-28	0.00E+00	1.01E-275	0.00E+00	0.00E+00
	best	6.15E-01	2.44E-136	7.76E-08	1.5E-118	1.95E-135	2.47E-81	0.00E+00	0.00E+00
f6	mean	4.30E+00	5.00E-133	1.56E-05	7.36E-04	3.74E-131	2.15E-77	**0.00E+00**	**0.00E+00**
	std	9.29E+00	1.88E-264	4.93E-10	1.72E-05	1.82E-260	1.88E-153	0.00E+00	0.00E+00
	best	5.19E-04	7.82E-05	9.84E-04	8.59E-05	1.35E-04	3.24E-03	4.19E-05	2.40E-05
f7	mean	4.08E-03	5.89E-04	7.36E-03	8.90E-04	5.71E-04	2.09E-02	4.77E-04	**3.71E-05**
	std	7.90E-06	5.85E-08	2.18E-05	6.72E-07	5.56E-08	1.81E-04	7.59E-08	1.87E-08
	best	1.62E-03	0.00E+00	6.87E-07	0.00E+00	0.00E+00	0.00E+00	7.22E-09	0.00E+00
f8	mean	2.30E-02	2.18E-26	5.54E-05	2.03E-01	4.40E-18	9.37E-05	1.50E-04	**0.00E+00**
	std	4.34E-04	2.38E-50	9.94E-09	4.07E-01	5.08E-34	4.39E-07	1.71E-07	0.00E+00
	best	4.04E-12	8.26E-264	1.26E-15	4.98E-23	8.64E-239	5.61E-170	9.38E-28	0.00E+00
f9	mean	1.98E-06	5.89E-256	1.98E-08	2.66E-10	3.64E-197	6.66E-159	6.62E-25	**0.00E+00**
	std	1.63E-11	0.00E+00	7.84E-15	1.19E-18	0.00E+00	0.00E+00	1.67E-48	0.00E+00
	best	2.58E-03	0.00E+00	5.18E-05	0.00E+00	0.00E+00	0.00E+00	6.40E-06	0.00E+00
f10	mean	6.22E-03	**0.00E+00**	5.44E-04	2.10E-13	**0.00E+00**	4.44E-18	1.77E-05	**0.00E+00**
	std	3.92E-06	0.00E+00	1.17E-07	2.19E-24	0.00E+00	9.86E-34	7.54E-11	0.00E+00
	best	1.27E-01	0.00E+00	4.80E-09	7.40E-03	3.39E-07	1.72E-02	0.00E+00	0.00E+00
f11	mean	3.01E-01	1.02E-05	3.76E-02	1.04E-01	1.99E-03	7.96E-02	7.77E-02	**0.00E+00**
	std	9.47E-03	5.21E-09	4.74E-04	5.90E-03	1.84E-05	1.82E-03	5.53E-03	0.00E+00
	best	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	mean	6.00E-02	**0.00E+00**	**0.00E+00**	1.80E-01	**0.00E+00**	**0.00E+00**	2.00E-02	**0.00E+00**
	std	5.76E-02	0.00E+00	0.00E+00	4.36E-01	0.00E+00	0.00E+00	2.00E-02	0.00E+00
	best	7.13E-03	0.00E+00	2.33E-06	1.99E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
f13	mean	1.05E-01	**0.00E+00**	1.92E-04	9.19E+00	7.21E-09	2.94E+00	1.51E+00	**0.00E+00**
	std	5.20E-03	0.00E+00	1.09E-07	2.85E+01	1.11E-15	3.02E+00	7.53E+00	0.00E+00
	best	2.73E-02	0.00E+00	4.88E-06	4.00E+00	2.69E-06	0.00E+00	0.00E+00	0.00E+00
f14	mean	9.45E-02	2.00E-02	1.21E-04	9.07E+00	4.03E-02	2.82E-02	1.02E+01	**0.00E+00**
	std	2.64E-03	2.00E-02	1.19E-08	1.22E+01	3.92E-02	1.02E-02	1.04E+02	0.00E+00
	best	7.30E-02	8.88E-16	1.73E-04	4.44E-15	8.88E-16	4.44E-15	8.88E-16	8.88E-16
f15	mean	2.56E-01	2.38E-15	9.30E-04	6.95E-02	8.39E-01	7.24E-02	4.51E-15	**8.88E-16**
	std	1.42E-02	3.14E-30	3.13E-07	7.68E-02	1.51E+01	2.62E-01	5.38E-29	0.00E+00
	best	1.88E-03	1.29E-159	2.20E-05	1.8E-103	5.80E-126	1.62E-91	1.32E-05	2.84E-221
f16	mean	4.70E-03	**2.11E-23**	1.23E-04	4.35E-02	2.15E-11	1.64E-03	4.38E-03	1.78E-16
	std	1.75E-06	9.03E-45	6.89E-09	3.70E-02	5.09E-21	8.54E-05	1.39E-04	7.62E-32
	best	1.36E+03	2.11E-134	1.15E-05	1.4E-116	7.88E-134	5.48E-79	0.00E+00	0.00E+00
f17	mean	2.57E+04	7.18E-131	7.94E-02	1.44E+01	5.15E-128	6.93E-75	**0.00E+00**	**0.00E+00**
	std	1.12E+09	3.42E-260	2.32E-02	8.09E+03	1.30E-253	3.92E-148	0.00E+00	0.00E+00
	best	3.00E-01	9.99E-02	2.00E-01	9.99E-02	9.99E-02	9.99E-02	2.00E-01	9.99E-02
f18	mean	6.01E-01	**9.99E-02**	2.35E-01	2.10E-01	2.85E-01	2.15E-01	3.65E-01	**9.99E-02**
	std	1.16E-02	1.45E-21	3.45E-03	1.57E-02	2.34E-02	5.55E-03	2.45E-02	8.25E-34

f19	best	2.38E-01	0.00E+00	3.32E-02	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	mean	4.28E-01	**0.00E+00**	1.19E-01	1.02E-02	**0.00E+00**	**0.00E+00**	4.26E-16	**0.00E+00**
	std	1.08E-02	0.00E+00	1.78E-03	4.23E-03	0.00E+00	0.00E+00	1.88E-30	0.00E+00
f20	best	3.15E-04	4.71E-32	8.46E-10	4.71E-32	4.71E-32	4.71E-32	1.22E-10	4.71E-32
	mean	9.37E-03	**4.71E-32**	1.51E-07	2.91E-02	**4.71E-32**	**4.71E-32**	6.11E-01	**4.71E-32**
	std	2.88E-04	2.75E-94	1.92E-13	1.06E-02	2.75E-94	2.75E-94	2.93E+00	2.75E-94
f21	best	2.11E-03	1.35E-32	1.51E-09	1.35E-32	1.35E-32	1.35E-32	2.88E-10	1.35E-32
	mean	1.59E-02	**1.35E-32**	8.07E-07	4.10E-03	**1.35E-32**	**1.35E-32**	2.50E-03	**1.35E-32**
	std	1.16E-04	1.22E-94	6.50E-12	6.19E-04	1.22E-94	1.22E-94	4.98E-05	8.03E-96
f22	best	3.12E-01	1.35E-31	2.19E-07	1.47E-31	1.35E-31	1.35E-31	1.35E-31	1.35E-31
	mean	1.41E+00	**1.35E-31**	1.27E-05	8.87E-01	1.81E+02	**1.35E-31**	3.62E+00	**1.35E-31**
	std	8.11E-01	3.13E-92	8.54E-10	1.37E+01	5.71E+05	3.13E-92	6.59E+01	0.00E+00
f23	best	1.14E+00	0.00E+00	7.41E-07	0.00E+00	0.00E+00	0.00E+00	1.07E-05	0.00E+00
	mean	2.84E+00	**0.00E+00**	1.61E-01	1.94E-01	**0.00E+00**	**0.00E+00**	1.07E-01	**0.00E+00**
	std	4.89E-01	0.00E+00	4.14E-01	1.47E+00	0.00E+00	0.00E+00	2.82E-01	0.00E+00
f24	best	1.37E-02	0.00E+00	3.81E-06	6.00E+00	1.68E-06	0.00E+00	3.64E-07	0.00E+00
	mean	1.16E-01	6.00E-02	1.07E-04	9.93E+00	2.04E-02	7.24E-03	6.00E-02	**0.00E+00**
	std	7.86E-03	5.76E-02	8.61E-09	1.04E+01	2.00E-02	5.95E-04	5.76E-02	0.00E+00

It can be seen from Table 9 that the optimization performance of the original BBO is not as good as that of the other 7 improved algorithms on the 10-dimensional test functions. Therefore, the research of other scholars and the algorithm proposed in this paper have improved the performance of the original BBO algorithm to some extent. Except for the original BBO, the other algorithms can converge to the optimal value accurately on at least one test function, and the standard deviation is 0. In contrast, the ZGBBO proposed in this paper has the best overall performance, which can accurately converge to the global optimal value every time on 15 test functions, such as single-peak functions f1, f3, f5, f6, f8-f10 and multi-modal functions f11-f14, f17, f19, f23, f24, etc. Although ZGBBO does not converge to the global optimal value every time on the test functions f4, f7, f15, f18 and f20-f22, its mean value and standard deviation are the best among the eight comparison algorithms, which showing better searching ability. It can be seen that the ZGBBO not only has high convergence accuracy in searching, but also is not easy to fall into local optimal state when solving multimodal problems. In addition, PRBBO, WRBBO, HBBO-CMA and DCGBBO all show strong competitiveness in the 10-dimensional search space. The mean value of PRBBO converges to the optimal value successfully on five problems, and the result is the best on one problem. On four problems, PRBBO and ZGBBO do not converge to the optimal value, but the obtained mean value is the same. WRBBO converges to the global optimal value on four problems, and the search results on two problems are consistent with ZGBBO. HBBO-CMA finds the global optimal value on three problems, and the mean value on five problems is the same as that of ZGBBO, which does not find the best solution. DCGBBO accurately finds the optimal result on five test questions, that is, the mean value and standard deviation obtained are all 0. Therefore, in the 10-dimensional optimization environment, the overall performance of ZGBBO algorithm is the best. It is not easy to fall into the local optimal solution, and the convergence accuracy is significantly higher than the other seven comparison algorithms.

Table 10 The results of eight BBO algorithms (30-demensional benchmark functions)

F	Index	BBO	PRBBO	BBOSB	TDBBO	WRBBO	HBBO-CMA	DCGBBO	ZGBBO
f1	best	1.50E-02	1.80E-50	5.01E-06	7.32E-66	2.34E-59	1.74E-28	4.30E-13	0.00E+00
	mean	3.90E-02	1.68E-49	3.61E-05	2.35E-62	1.41E-58	1.93E-27	3.42E-12	**0.00E+00**
	std	2.54E-04	9.05E-99	6.80E-10	1.0E-123	7.41E-117	5.48E-54	7.82E-24	0.00E+00
f2	best	9.49E-02	2.78E-31	1.31E-03	4.89E-34	1.59E-32	2.28E-17	1.87E-08	8.79E-182
	mean	1.90E-01	5.96E-31	3.27E-03	1.64E-33	4.30E-32	1.18E-16	8.91E-08	**2.84E-179**
	std	1.61E-03	4.93E-62	1.45E-06	9.76E-67	3.68E-64	5.53E-33	3.81E-15	0.00E+00
f3	best	9.16E-02	4.12E-49	5.79E-07	6.45E-66	1.91E-58	1.39E-27	3.77E-12	0.00E+00
	mean	2.20E-01	1.78E-48	2.67E-06	7.39E-62	2.02E-57	1.41E-26	2.59E-11	**0.00E+00**
	std	6.13E-03	1.40E-96	4.44E-12	5.9E-122	2.57E-114	1.21E-52	6.36E-22	0.00E+00
f4	best	1.33E+00	5.22E-08	1.69E-02	1.46E+00	1.04E-11	1.29E-04	2.32E+00	5.05E-90
	mean	1.96E+00	8.55E-08	2.75E-02	4.77E+00	3.57E-10	8.92E-04	3.19E+01	**3.68E-89**
	std	1.40E-01	4.90E-16	2.98E-05	2.49E+00	8.10E-19	1.20E-06	3.03E+02	1.29E-177
f5	best	1.19E+02	1.71E-77	2.23E-07	9.9E-120	1.39E-102	1.83E-44	1.50E-15	0.00E+00
	mean	1.43E+03	6.93E-76	9.09E-06	1.6E-111	1.73E-100	2.33E-41	2.90E-13	**0.00E+00**
	std	8.85E+05	8.54E-151	6.92E-11	7.6E-221	1.33E-199	1.72E-81	2.55E-25	0.00E+00
f6	best	1.25E+01	3.39E-47	1.34E-05	6.18E-63	1.69E-56	1.17E-25	2.22E-10	0.00E+00
	mean	3.64E+01	1.18E-46	3.91E-04	4.44E-60	7.98E-56	1.50E-24	2.82E-09	**0.00E+00**
	std	2.91E+02	4.90E-93	1.10E-07	1.2E-118	3.68E-111	1.46E-48	6.98E-18	0.00E+00
f7	best	1.67E-03	1.42E-03	1.12E-02	9.99E-04	1.28E-03	2.74E-02	7.35E-04	1.65E-04
	mean	1.08E-02	3.32E-03	2.97E-02	2.84E-03	2.27E-03	9.53E-02	1.79E-03	**6.53E-04**
	std	2.17E-05	9.89E-07	1.17E-04	7.10E-07	3.24E-07	1.37E-03	4.85E-07	6.33E-08
f8	best	2.98E-02	2.44E-04	3.96E-05	1.48E-31	0.00E+00	2.01E-27	2.74E-10	0.00E+00
	mean	8.46E-02	1.24E-02	2.81E-04	7.98E-02	1.50E-04	5.86E-03	1.62E+00	**5.20E-30**
	std	1.45E-03	1.15E-04	3.30E-08	1.71E-01	8.54E-07	1.72E-03	1.31E+02	4.75E-59

f9	best	2.18E-11	1.45E-128	2.08E-19	1.13E-39	5.17E-145	7.63E-101	1.37E-22	0.00E+00
	mean	2.99E-07	3.11E-124	1.45E-09	5.83E-22	2.00E-127	5.30E-93	7.35E-20	**0.00E+00**
	std	2.58E-13	6.36E-247	2.00E-17	1.51E-41	1.48E-252	2.29E-184	2.47E-38	0.00E+00
f10	best	7.54E-03	2.22E-16	5.02E-04	2.22E-16	0.00E+00	2.22E-16	4.94E-03	0.00E+00
	mean	1.16E-02	2.22E-16	1.34E-03	2.31E-16	2.13E-16	2.62E-16	7.42E-03	**0.00E+00**
	std	3.61E-06	0.00E+00	2.17E-07	1.93E-33	1.93E-33	7.43E-33	1.33E-06	0.00E+00
f11	best	1.41E-01	0.00E+00	1.16E-08	0.00E+00	0.00E+00	0.00E+00	3.11E-12	0.00E+00
	mean	3.31E-01	**0.00E+00**	1.43E-03	1.38E-03	1.48E-04	7.82E-03	8.85E-03	**0.00E+00**
	std	8.00E-03	0.00E+00	1.17E-05	1.27E-05	1.09E-06	1.64E-04	2.15E-04	0.00E+00
f12	best	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	mean	2.00E-02	**0.00E+00**	**0.00E+00**	1.00E-01	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
	std	2.00E-02	0.00E+00	0.00E+00	1.33E-01	0.00E+00	0.00E+00	0.00E+00	0.00E+00
f13	best	3.80E-02	5.20E-10	8.03E-05	1.29E+01	2.20E+00	8.96E+00	3.77E-09	0.00E+00
	mean	1.17E-01	2.87E-03	5.60E-04	3.70E+01	6.35E+00	2.22E+01	2.37E+00	**0.00E+00**
	std	2.08E-03	7.54E-05	1.79E-07	1.86E+02	3.78E+00	4.96E+01	5.81E+00	0.00E+00
f14	best	5.21E-02	6.91E+00	1.07E-04	2.10E+01	1.82E+01	0.00E+00	1.62E-10	0.00E+00
	mean	1.18E-01	9.10E+00	5.39E-04	5.50E+01	2.15E+01	3.17E-01	5.60E-01	**0.00E+00**
	std	2.52E-03	5.32E-01	1.39E-07	3.15E+02	2.20E+00	5.21E-01	1.19E+00	0.00E+00
f15	best	9.18E-02	4.44E-15	4.04E-04	4.44E-15	4.95E+00	1.15E-14	3.35E-07	4.44E-15
	mean	1.51E-01	**4.44E-15**	1.13E-03	6.71E-15	1.88E+01	1.93E+00	1.20E-06	**4.44E-15**
	std	1.52E-03	0.00E+00	1.08E-07	2.97E-30	1.41E+01	3.43E+01	3.96E-13	0.00E+00
f16	best	5.43E-03	3.92E-07	1.05E-04	1.79E-34	1.38E-26	2.22E-15	5.14E-06	5.13E-181
	mean	9.00E-03	1.45E-04	3.64E-04	3.66E-05	3.30E-07	2.95E-02	8.89E-02	**2.04E-101**
	std	2.75E-06	6.00E-09	2.41E-08	6.00E-08	4.59E-12	1.73E-02	3.94E-01	3.46E-201
f17	best	2.23E+03	4.51E-46	8.71E-04	5.41E-62	5.34E-56	1.32E-24	3.21E-09	0.00E+00
	mean	1.54E+04	1.82E-45	5.57E-02	7.14E-58	2.97E-55	2.00E-23	4.43E-08	**3.35E-305**
	std	2.06E+08	1.05E-90	3.85E-03	3.94E-114	5.47E-110	4.64E-46	5.54E-15	0.00E+00
f18	best	7.00E-01	9.99E-02	3.00E-01	3.00E-01	2.00E-01	4.00E-01	4.00E-01	9.99E-02
	mean	1.31E+00	1.39E-01	5.75E-01	3.65E-01	4.95E-01	7.10E-01	1.57E+00	**1.10E-01**
	std	5.93E-02	1.94E-03	1.78E-02	2.39E-03	7.52E-02	2.73E-02	5.58E-01	1.00E-03
f19	best	7.29E-01	0.00E+00	1.52E-01	0.00E+00	0.00E+00	0.00E+00	3.48E-06	0.00E+00
	mean	9.75E-01	**0.00E+00**	3.16E-01	3.34E-04	**0.00E+00**	**0.00E+00**	4.13E-05	**0.00E+00**
	std	1.43E-02	0.00E+00	6.29E-03	5.42E-06	0.00E+00	0.00E+00	4.70E-10	0.00E+00
f20	best	2.27E-04	1.57E-32	3.60E-09	1.57E-32	1.57E-32	9.20E-30	1.74E-12	1.57E-32
	mean	2.26E-03	**1.57E-32**	3.65E-08	2.07E-03	**1.57E-32**	8.29E-03	2.68E-01	1.94E-32
	std	6.63E-06	3.06E-95	2.56E-15	2.15E-04	3.06E-95	8.07E-04	3.20E-01	4.09E-65
f21	best	3.62E-03	1.35E-32	4.37E-08	1.47E-32	1.35E-32	5.06E-29	5.46E-12	1.35E-32
	mean	1.48E-02	**1.35E-32**	9.07E-07	8.79E-04	**1.35E-32**	4.39E-04	2.20E-04	**1.35E-32**
	std	6.63E-05	1.22E-94	4.94E-12	9.07E-06	1.22E-94	4.73E-06	2.41E-06	8.32E-96
f22	best	7.28E-01	1.35E-31	1.81E-06	1.60E-31	1.35E-31	1.44E-16	2.86E-22	1.35E-31
	mean	1.55E+00	**1.35E-31**	5.48E-06	9.07E-04	1.60E+04	1.97E+01	4.82E+02	**1.35E-31**
	std	3.13E-01	3.13E-92	1.42E-11	1.65E-05	8.24E+08	4.18E+03	5.16E+05	0.00E+00
f23	best	3.50E+00	0.00E+00	2.78E-05	0.00E+00	0.00E+00	0.00E+00	2.09E+00	0.00E+00
	mean	6.88E+00	**0.00E+00**	8.83E-05	2.22E-18	**0.00E+00**	**0.00E+00**	5.96E+00	**0.00E+00**
	std	1.92E+00	0.00E+00	2.67E-09	2.47E-34	0.00E+00	0.00E+00	2.58E+00	0.00E+00
f24	best	3.61E-02	7.83E+00	5.69E-05	2.10E+01	1.73E+01	0.00E+00	1.40E+01	0.00E+00
	mean	1.12E-01	9.10E+00	4.90E-04	5.24E+01	2.11E+01	3.17E-01	1.86E+01	**0.00E+00**
	std	2.03E-03	4.01E-01	1.20E-07	3.61E+02	2.11E+00	2.06E+00	4.11E+00	0.00E+00

It can be seen from Table 10 that the ZGBBO algorithm proposed in this paper gets the best mean value among the eight algorithms on 23 test functions in a 30-dimensional search space, so the overall performance is still better than the other seven BBO algorithms. With the increase of the dimension, the searching ability of the other seven comparison algorithms decreases to varying degrees, and only ZGBBO still maintains excellent search performance. For example, DCGBBO can accurately search the global optimal value on the test functions f1, f2, f5, f6 and f17 in the 10-dimensional search space, the mean value and standard deviation are all 0. However, in the 30-dimensional search environment, DCGBBO's optimization performance decreases significantly, and the convergence accuracy decreases rapidly, so the algorithm is not suitable for the problem environment with higher dimensions. In contrast, ZGBBO algorithm shows obvious advantages over other seven comparison algorithms in terms of mean values for single-peak problems f1-f10. For multimodal problems, on all functions, ZGBBO converges to the global optimal value or the result is the best except the mean obtained on problem f20 is not as accurate as PRBBO and WRBBO. Especially for functions f2, f4, f16 and f17, the mean obtained by ZGBBO is at least 100 exponential levels higher than other algorithms. Besides, PRBBO also shows well competitiveness. For instance, with the mean value of seven test functions being the same as that of ZGBBO, and the result of one test function being better than that of ZGBBO. According to the above analysis, ZGBBO algorithm can still maintain outstanding search performance in the 30-dimensional optimization environment, and is more effective than other comparison algorithms.

Table 11 The results of eight BBO algorithms (50-demensional benchmark functions)

F	Index	BBO	PRBBO	BBOSB	TDBBO	WRBBO	HBBO-CMA	DCGBBO	ZGBBO
	best	7.74E-02	5.69E-28	1.37E-03	1.20E-46	2.23E-38	7.76E-18	1.42E-06	3.16E-298
f1	mean	1.31E-01	2.59E-27	3.05E-03	8.00E-44	6.03E-38	2.79E-17	5.32E-06	**8.65E-295**
	std	1.58E-03	1.15E-54	8.14E-07	1.65E-85	7.86E-76	3.56E-34	3.35E-12	0.00E+00
	best	2.19E-01	6.62E-18	1.93E-02	8.08E-26	1.63E-20	6.38E-11	4.45E-04	6.52E-171
f2	mean	3.26E-01	9.78E-18	3.05E-02	3.49E-25	3.76E-20	1.27E-10	7.47E-04	**1.06E-169**
	std	2.43E-03	4.22E-36	2.79E-05	3.42E-50	1.75E-40	1.93E-21	2.63E-08	0.00E+00
	best	2.01E-01	1.28E-27	7.51E-05	4.00E-47	2.38E-38	9.99E-18	4.51E-06	2.23E-299
f3	mean	5.21E-01	3.06E-27	1.36E-04	8.34E-45	6.99E-38	6.52E-17	1.58E-05	**1.19E-295**
	std	2.78E-02	1.48E-54	2.28E-09	3.36E-88	1.08E-75	2.02E-33	4.34E-11	0.00E+00
	best	2.68E+00	3.54E-03	8.01E-02	3.86E+00	3.31E-05	5.21E-01	3.41E+01	3.69E-79
f4	mean	3.64E+00	4.69E-03	1.14E-01	7.52E+00	8.76E-05	2.22E+00	4.84E+01	**6.19E-78**
	std	3.25E-01	5.35E-07	1.78E-04	3.77E+00	3.76E-09	1.80E+00	5.76E+01	4.90E-155
	best	1.53E+04	9.24E-35	1.54E-02	5.57E-81	3.57E-60	1.85E-21	1.31E-04	0.00E+00
f5	mean	9.15E+04	1.39E-33	9.09E-02	1.48E-75	4.51E-59	1.51E+07	2.91E-03	**0.00E+00**
	std	4.11E+09	1.96E-66	2.93E-03	2.28E-149	3.60E-117	6.08E+15	7.99E-06	0.00E+00
	best	7.00E+01	9.44E-25	1.68E-02	5.32E-44	1.92E-35	3.78E-15	1.69E-03	3.03E-294
f6	mean	2.05E+02	2.96E-24	4.26E-02	2.84E-41	4.98E-35	3.52E-14	6.42E-03	**3.91E-291**
	std	6.62E+03	1.85E-48	1.95E-04	4.10E-81	4.09E-70	8.39E-28	6.56E-06	0.00E+00
	best	1.18E-02	3.74E-03	4.52E-02	4.76E-03	3.62E-03	7.92E-02	2.31E-03	2.16E-04
f7	mean	2.53E-02	6.77E-03	8.44E-02	7.44E-03	5.13E-03	1.81E-01	4.80E-03	**6.33E-04**
	std	5.52E-05	3.22E-06	3.87E-04	2.90E-06	6.21E-07	4.63E-03	3.36E-06	9.66E-08
	best	1.22E-01	2.51E+00	8.74E-03	7.78E-30	5.76E-27	9.44E-17	3.85E-05	6.09E-18
f8	mean	3.21E-01	3.86E+00	2.18E-02	6.58E-02	6.35E-03	1.42E-02	1.13E+01	**7.28E-16**
	std	1.24E-02	4.53E-01	6.88E-05	2.16E-01	1.87E-03	1.01E-02	8.06E+02	6.34E-30
	best	6.88E-12	8.95E-83	1.98E-19	6.64E-46	1.10E-108	4.14E-80	3.43E-20	0.00E+00
f9	mean	1.61E-07	1.10E-80	1.63E-09	5.06E-36	1.19E-100	3.80E-69	9.53E-17	**0.00E+00**
	std	4.39E-14	7.54E-160	3.21E-17	3.96E-70	3.17E-199	5.17E-136	3.96E-32	0.00E+00
	best	1.69E-02	2.22E-16	7.23E-03	2.22E-16	2.22E-16	3.84E-12	6.19E-02	0.00E+00
f10	mean	2.21E-02	2.22E-16	1.20E-02	3.86E-16	2.22E-16	8.76E-12	8.48E-02	**1.18E-16**
	std	8.24E-06	0.00E+00	6.49E-06	9.68E-33	0.00E+00	1.07E-23	1.51E-04	1.27E-32
	best	3.05E-01	0.00E+00	1.63E-06	0.00E+00	0.00E+00	0.00E+00	1.46E-05	0.00E+00
f11	mean	4.43E-01	**0.00E+00**	1.83E-03	8.38E-04	1.18E-10	2.17E-03	8.17E-03	**0.00E+00**
	std	7.06E-03	0.00E+00	1.73E-05	6.60E-06	6.98E-19	1.65E-05	8.61E-05	0.00E+00
	best	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	mean	6.00E-02	**0.00E+00**	**0.00E+00**	1.80E-01	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
	std	5.76E-02	0.00E+00	0.00E+00	2.73E-01	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	best	1.02E-01	1.72E+01	8.95E-03	2.49E+01	2.98E+01	3.18E+01	6.97E+00	0.00E+00
f13	mean	2.51E-01	2.17E+01	2.56E-02	6.35E+01	4.64E+01	5.14E+01	1.51E+01	**0.00E+00**
	std	8.22E-03	3.32E+00	6.35E-05	3.88E+02	3.29E+01	1.28E+02	1.57E+01	0.00E+00
	best	1.04E-01	2.64E+01	7.81E-03	4.80E+01	4.45E+01	2.05E-07	5.00E+00	0.00E+00
f14	mean	2.47E-01	2.89E+01	2.37E-02	1.07E+02	5.41E+01	1.26E+00	1.27E+01	**0.00E+00**
	std	5.30E-03	9.86E-01	8.35E-05	1.42E+03	1.31E+01	4.39E+00	9.06E+00	0.00E+00
	best	1.19E-01	1.87E-14	4.52E-03	7.99E-15	2.00E+01	1.12E-09	5.06E-04	4.44E-15
f15	mean	1.79E-01	2.50E-14	6.73E-03	9.06E-15	2.01E+01	9.08E+00	8.56E-04	**4.44E-15**
	std	1.35E-03	1.72E-29	1.25E-06	3.74E-30	1.19E-03	9.88E+01	7.09E-08	0.00E+00
	best	1.08E-02	3.89E-03	2.07E-03	3.29E-19	6.46E-12	8.03E-08	8.82E-04	9.12E-172
f16	mean	1.64E-02	4.59E-03	3.13E-03	1.45E-05	3.38E-05	4.96E-02	2.68E-01	**1.62E-124**
	std	5.47E-06	1.35E-07	5.15E-07	1.06E-08	1.60E-08	3.51E-02	1.13E+00	3.21E-247
	best	3.09E+03	5.16E-24	1.37E-01	5.32E-42	2.85E-35	7.65E-14	1.53E-02	6.74E-294
f17	mean	2.30E+04	1.48E-23	5.44E-01	4.55E-40	7.56E-35	3.41E-13	6.41E-02	**6.93E-290**
	std	2.31E+08	5.72E-47	1.17E-01	6.19E-79	7.49E-70	7.68E-26	2.71E-03	0.00E+00
	best	1.60E+00	2.00E-01	8.00E-01	4.00E-01	9.99E-02	1.00E+00	7.00E-01	9.99E-02
f18	mean	1.94E+00	2.00E-01	8.95E-01	4.95E-01	5.80E-01	1.40E+00	3.11E+00	**1.33E-01**
	std	4.89E-02	1.28E-11	9.97E-03	8.92E-03	9.54E-02	1.07E-01	4.46E+00	2.30E-03
	best	1.22E+00	0.00E+00	9.57E-01	0.00E+00	0.00E+00	6.55E-11	1.85E-02	0.00E+00
f19	mean	1.67E+00	**0.00E+00**	1.38E+00	3.78E-03	**0.00E+00**	2.75E-02	2.93E-02	**0.00E+00**
	std	3.11E-02	0.00E+00	4.90E-02	3.31E-04	0.00E+00	1.72E-02	2.63E-05	0.00E+00
	best	3.53E-04	6.56E-28	2.82E-07	9.42E-33	9.42E-33	5.83E-19	4.19E-08	7.71E-22
f20	mean	1.54E-03	2.39E-27	8.17E-07	7.46E-03	**9.42E-33**	2.24E-02	6.26E-02	1.07E-14
	std	3.58E-06	1.85E-54	8.76E-14	5.75E-04	1.91E-96	5.03E-03	1.72E-02	3.42E-27
	best	1.06E-02	6.73E-27	6.88E-06	2.21E-32	1.35E-32	6.39E-18	6.39E-07	9.02E-21
f21	mean	2.39E-02	1.85E-26	2.00E-05	1.52E-03	**1.35E-32**	1.54E-03	1.98E-06	3.30E-03
	std	4.33E-05	7.70E-53	5.20E-11	1.90E-05	1.22E-94	4.44E-05	5.35E-13	3.26E-04
	best	1.51E+00	4.22E-23	4.82E-05	2.95E-31	2.09E+01	1.15E-01	4.72E+02	4.67E-17
f22	mean	3.42E+00	**1.05E-22**	9.20E-05	1.04E+00	4.52E+04	3.40E+03	1.73E+04	4.59E-01
	std	1.10E+00	3.07E-45	1.18E-09	1.29E+01	2.99E+09	2.83E+07	2.22E+08	5.15E-01
	best	7.75E+00	0.00E+00	2.56E-03	0.00E+00	0.00E+00	1.05E-15	2.77E+01	0.00E+00
f23	mean	1.23E+01	**0.00E+00**	6.39E-03	8.78E-02	**0.00E+00**	5.51E-14	4.34E+01	**0.00E+00**
	std	3.56E+00	0.00E+00	6.66E-06	7.27E-02	0.00E+00	5.46E-26	6.34E+01	0.00E+00
	best	1.19E-01	2.60E+01	1.08E-02	4.30E+01	4.57E+01	1.76E-07	5.50E+01	0.00E+00
f24	mean	2.43E-01	2.89E+01	2.48E-02	8.98E+01	5.29E+01	6.84E-02	1.04E+02	**0.00E+00**
	std	4.93E-03	1.60E+00	6.29E-05	1.08E+03	8.68E+00	1.54E+00	5.97E+02	0.00E+00

As can be seen from Table 11, in the 50-dimensional problem environment, ZGBBO still has the best overall performance among the eight comparison algorithms. Obviously, ZGBBO shows an absolute advantage on the single-peak functions f1-f10, and the mean value obtained on all the single-peak functions is the smallest among the eight algorithms. This is because the convex migration mechanism and example learning approach proposed in this paper can effectively improve the speed of the population moving to the global optimal solution, so that the algorithm can quickly converge. On the multi-modal functions, the performance of ZGBBO on the functions f20-f22 decreases with the increase of the problem dimension, and the result obtained is not as desired as WRBBO and PRBBO. However, for other multi-modal problems, the search results of ZGBBO are the best among the eight algorithms, such as the multi-modal functions f11-f14, f19, f23 and f24, ZGBBO converges to the global optimal value accurately, and the mean and standard deviation are all 0, which showing excellent search performance. Therefore, on the 50-dimension problems, ZGBBO maintains good overall performance, and its search ability basically does not decrease with the increase of dimension, which has good robustness.

From the overall observation of Tables 9-11, it can be found that the optimization performance of ZGBBO algorithm is less sensitive to the problem dimension, and it can still maintain excellent searching ability on high-dimensional problems. For the single-peak problems, ZGBBO algorithm can always move to the global optimal solution quickly in high dimensional environment, and keep high calculation accuracy. On multi-modal problems, ZGBBO algorithm can always successfully jump out of the local optimal state and converge to the global optimal value in high dimensional environment. The other seven BBO variants, such as PRBBO, WRBBO and DCGBBO, have strong competitiveness in the 10-dimensional problem environment, and the overall performance is wonderful. However, with the problem dimension increasing, only PRBBO maintains better competitiveness in the 30-dimensional problem environment. In the 50-dimensional problem environment, The performance of these algorithms reduces quickly, and is not as stable as ZGBBO algorithm. The convergence accuracy of ZGBBO algorithm decreases slightly with the increase of problem dimension. Especially for the functions f5, f9, f11-f14, f19, f23 and f24, the convergence accuracy of ZGBBO algorithm is not affected by the problem dimension, and it can still find the global optimal value precisely. However, the convergence precision of the other seven comparison algorithms decreases greatly with the increase of the problem dimension, so these algorithms do not have ideal robustness and are not suitable for high-dimensional problems. With the development of modern society, the requirements of algorithms to solve practical problems in life have gradually increased, and algorithms must be able to solve high-dimensional problems effectively. The ZGBBO algorithm proposed in this paper maintains excellent optimization performance in high dimensions. Therefore, ZGBBO algorithm is more advanced and effective, which is worth adopting and promoting.

In order to fully compare the overall performance of the eight BBO algorithms, Friedman test is performed on the eight algorithms according to the optimization results of the algorithms in Table 11 in a 50-dimensional search environment. The specific results are shown in Table 12 and Fig. 7.

It can be seen from Table 12 that the search results of ZGBBO algorithm on 21 test functions are the best among eight BBO algorithms, and the search results only on functions f20-f22 are not as ideal as other algorithms. According to the average ranking of eight algorithms in Fig. 7, PRBBO and WRBBO are two BBO variant algorithms with strong competitiveness, which are effective and advanced among BBO improved algorithms. However, the DCGBBO algorithm proposed in 2021 is only better than BBO algorithm in the search results, so its competitiveness is weak. According to Table 12 and Fig. 7, ZGBBO ranks the highest in average and the first in total among the eight comparison algorithms, which has the best overall performance. Therefore, ZGBBO is an advanced algorithm with stron g competitiveness and high optimization performance.

Table 12 The Friedman test of eight BBO algorithms according to the Table 11

	BBO	PRBBO	BBOSB	TDBBO	WRBBO	HBBO-CMA	DCGBBO	ZGBBO
f1	8	4	7	2	3	5	6	1
f2	8	4	7	2	3	5	6	1
f3	8	4	7	2	3	5	6	1
f4	6	3	4	7	2	5	8	1
f5	7	4	6	2	3	8	5	1
f6	8	4	7	2	3	5	6	1
f7	6	4	7	5	3	8	2	1

f8	6	7	4	5	2	3	8	1
f9	8	3	7	5	2	4	6	1
f10	7	2	6	4	2	5	8	1
f11	8	2	5	4	3	6	7	1
f12	7	1	1	8	1	1	1	1
f13	3	5	2	8	6	7	4	1
f14	3	6	2	8	7	4	5	1
f15	6	3	5	2	8	7	4	1
f16	6	5	4	2	3	7	8	1
f17	8	4	7	2	3	5	6	1
f18	7	2	5	3	4	6	8	1
f19	8	1	7	4	1	5	6	1
f20	5	2	4	6	1	7	8	3
f21	8	2	4	5	1	6	3	7
f22	5	1	2	4	8	6	7	3
f23	7	1	5	6	1	4	8	1
f24	3	5	2	7	6	4	8	1
Mean Rank	6.50	3.2917	4.88	4.38	3.2917	5.33	6.00	1.42
Final Rank	8	2	5	4	2	6	7	1

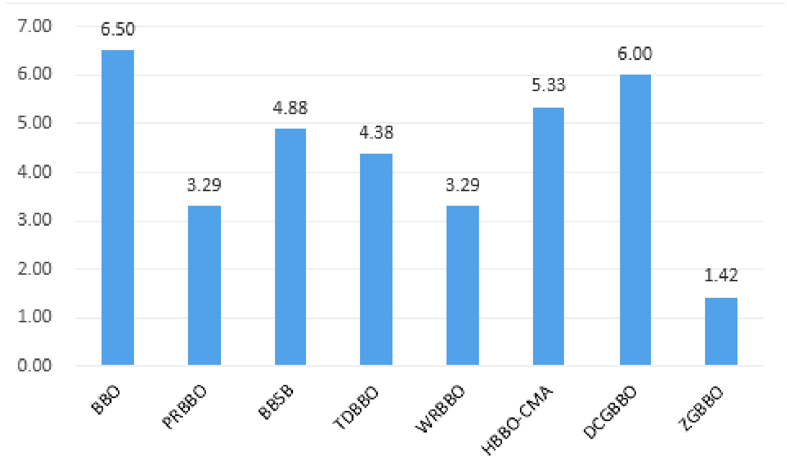
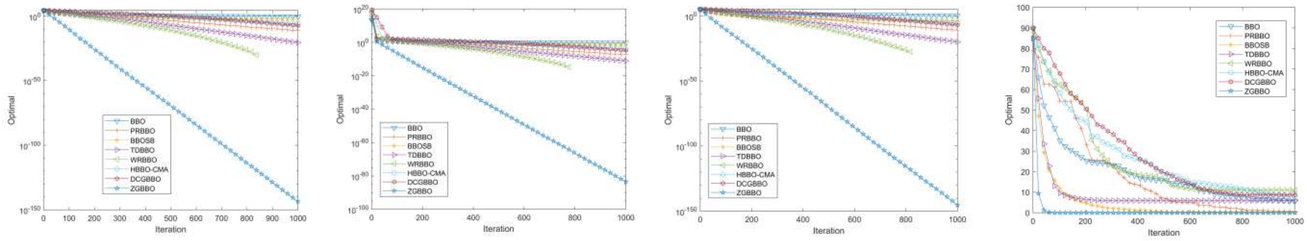


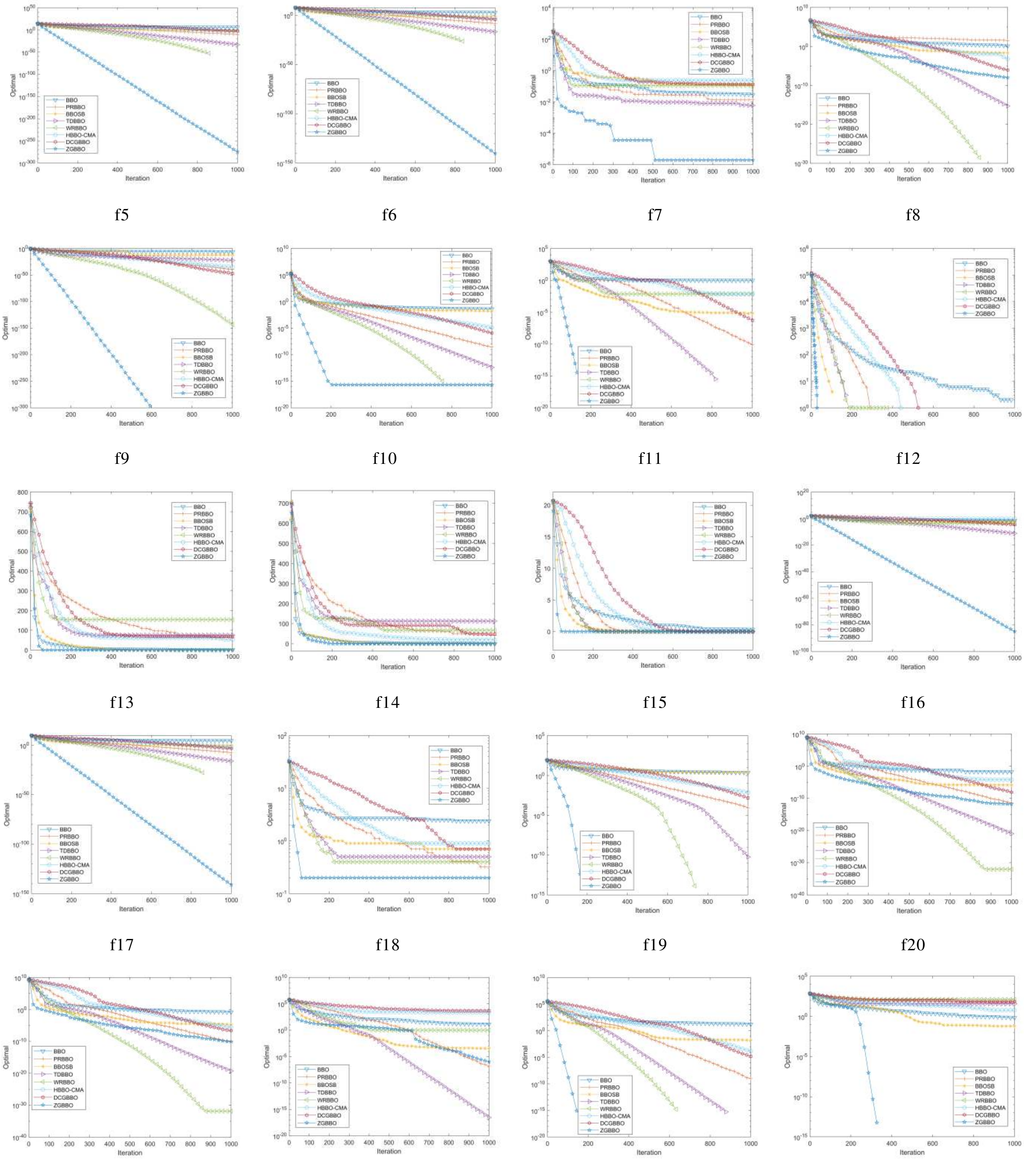
Fig. 7 Average ranking of eight BBO algorithms

5.3.3 Comparison of convergence rates

The experiment in this section is to compare the convergence speed of ZGBBO and other seven BBO variant algorithms on 24 different test functions in 50 dimensions, so as to compare the algorithm performance. The specific operation is to set the maximum number of iterations T=1000 and the population size NP=5D. Each algorithm is run independently for 50 times. The optimal running result and the search results of each generation are taken to plot the optimal convergence images of the eight algorithms on different test functions, as shown in Fig. 8.



f1 f2 f3 f4



f5 f6 f7 f8

f9 f10 f11 f12

f13 f14 f15 f16

f17 f18 f19 f20

f21 f22 f23 f24

Fig. 8 Convergence of eight BBO algorithms on different test functions

It can be observed from the 24 sub-graphs in Fig. 8, the convergence speed of the ZGBBO algorithm proposed in this paper is significantly faster than that of other seven comparison algorithms on 20 test functions, which showing obvious advantages. On the single-peak functions f1-f7, f9 and f10, ZGBBO rapidly converges to the global optimal solution from the beginning of iteration, and the convergence precision is 10 to 150 exponential levels higher than other algorithms. On the function f8, ZGBBO's convergence speed is not as fast as TDBBO and WRBBO, but it is better than the other five algorithms. Therefore, ZGBBO is more competitive than other BBO algorithms with higher search efficiency and rapider convergence rate on single-peak problems. On the multi-modal problems, ZGBBO also reflects a huge advantage. On the functions f11-f19, f23 and f24, the convergence rate of ZGBBO is significantly faster than that of the other seven BBO variants. Especially on f11, f12, f19, f23 and f24, the convergence curve of ZGBBO is almost invisible, and it rapidly approaches to the optimal value from the beginning of evolution. It can also be seen from Fig. 8 that on the multi-modal functions f11, f16-f19, f23 and f24, the other seven comparison algorithms fall into local optimal solution, resulting in search stagnation, while ZGBBO does not fall into local optimal value, and the algorithm is always in convergence state. Therefore, ZGBBO can effectively jump out of the local optimal solution, and the overall performance is better than other congeneric algorithms. Although ZGBBO is not the fastest convergent on f20-f22, it is only slower than WRBBO or TDBBO. In contrast, WRBBO and TDBBO show strong competitiveness in algorithm convergence speed. It is impossible for an algorithm to show optimal performance on all problems, so ZGBBO is still an algorithm worth adopting and developing.

5.4 Comparison between ZGBBO and other evolutionary algorithms

In this section, we compare the performance of ZGBBO with six state-of-the-art evolutionary algorithms proposed in the resent three years. Six highly advanced and competitive new population intelligence optimization algorithms are selected for comparison, they are electrostatic discharge algorithm (ESDA) (Houssem et al. 2019), fitness dependent optimizer (FDO) (Abdullah and Rashid 2019), artificial electric field algorithm (AEFA) (Anita and Yadav 2019), chimpanzee optimization algorithm (ChOA) (Khishe and Mosavi 2020), marine predator algorithm (MPA) (Faramarzi et al. 2020), archimedes optimization algorithm (AOA) (Hashim et al. 2021). The parameter settings of the six state-of-the-art evolutionary algorithms are consistent with their original references, as shown in Table 13.

Table 13 Parameter setting of six state-of-the-art algorithms

Article	Time	Algorith	Parameter setting
(Houssem et al. 2019)	2019	ESDA	$\mu=0.7,\sigma=0.2,\beta_1\sim N(\mu,\sigma),\beta_2\sim N(\mu,\sigma),\beta_3\sim N(\mu,\sigma)$
(Abdullah and Rashid 2019)	2019	FDO	$r=unifrnd[-1,1], fw\in[0,1]$
(Anita and Yadav 2019)	2019	AEFA	$K_0=500,\alpha=30$
(Khishe and Mosavi 2020)	2020	ChOA	$r_1=rand,r_2=rand,c=2r_2,m=Chaotic\_value$ (Chaos vector)
(Faramarzi et al. 2020)	2020	MPA	$P=0.5,R=rand,r=rand,FADs=0.2$
(Hashim et al. 2021)	2021	AOA	$C_1=2,C_2=6,C_3=2,C_4=0.5$

5.4.1 Optimize performance comparison

Consistent with section 5.3, seven algorithms search for optimization on 24 test functions in Table 3, set the maximum evaluation times (FEs) of each test function, and record the optimal value searched when FEs is reached. The FEs of each test function is also the same as section 5.3. Each algorithm searches for the optimal values of 24 test functions in the search space of *D*=10, *D*=30 and *D*=50 respectively, *D* is the problem dimension. Similarly, in order to avoid contingency, each algorithm is independently run for 50 times on different problem dimensions, and the best result (Best), mean value (Mean) and standard deviation (Std) among the 50 results as evaluation indexes, among which the mean value is the focus of comparison. Tables 14-16 show the optimization results of seven evolutionary algorithms on 24 test functions in 10 dimensions, 30 dimensions and 50 dimensions respectively, where the best results of the seven algorithms are shown in bold.

Table 14 The results of ZGBBO and six state-of-the-art algorithms (10-demensional benchmark functions)

F	Index	ESDA	FDO	AEFA	ChOA	MPA	AOA	ZGBBO
	best	1.80E-16	4.00E-158	1.23E-24	1.55E-192	2.02E-136	2.81E-70	0.00E+00
f1	mean	8.03E-07	5.58E-155	2.40E-24	3.25E-117	4.25E-131	9.64E-58	**0.00E+00**
	std	1.84E-06	0.00E+00	8.94E-25	1.78E-116	1.78E-130	2.43E-57	0.00E+00
	best	1.42E-08	8.73E-113	9.47E-13	2.11E-113	2.97E-76	1.20E-36	4.08E-222
f2	mean	6.86E-04	2.04E-109	1.42E-12	9.05E-76	2.06E-70	1.71E-30	**4.49E-219**
	std	1.36E-03	3.70E-109	2.15E-13	4.85E-75	7.62E-70	3.84E-30	0.00E+00
	best	1.21E-10	4.92E-161	1.03E-25	1.22E-206	8.80E-140	3.79E-88	0.00E+00
f3	mean	4.59E-05	7.59E-158	3.04E-25	6.89E-126	2.59E-133	8.05E-62	**0.00E+00**
	std	1.58E-04	0.00E+00	9.38E-26	3.77E-125	1.01E-132	3.60E-61	0.00E+00
	best	4.38E-05	1.73E-64	1.70E-13	6.73E-76	9.33E-56	3.70E-17	2.91E-142
f4	mean	2.08E-03	5.35E-64	3.50E-13	2.17E-32	2.31E-53	1.39E-14	**8.72E-137**
	std	2.07E-03	4.61E-64	6.11E-14	1.12E-31	8.85E-53	5.00E-14	1.06E-271
	best	1.13E-23	3.03E-184	4.41E-46	0.00E+00	3.09E-252	2.52E-130	0.00E+00
f5	mean	2.04E-07	2.05E-177	1.90E-21	3.58E-220	1.34E-240	8.03E-100	**0.00E+00**
	std	6.69E-07	0.00E+00	8.49E-21	0.00E+00	0.00E+00	2.86E-99	0.00E+00
	best	2.45E-12	2.18E-154	1.64E-09	1.82E-194	6.93E-134	4.88E-67	0.00E+00
f6	mean	2.18E-03	1.61E-152	2.55E-02	3.76E-116	1.09E-128	2.30E-54	**0.00E+00**
	std	3.92E-03	2.41E-152	5.71E-02	2.06E-115	2.56E-128	1.02E-53	0.00E+00

f7	best	1.42E-05	3.38E-01	8.56E-04	1.57E-07	6.16E-06	1.74E-04	2.40E-05
	mean	3.99E-04	5.94E-01	2.52E-03	**1.11E-06**	1.20E-04	6.24E-04	3.71E-05
	std	2.40E-04	3.00E-01	1.01E-03	9.66E-06	7.56E-05	3.89E-04	1.87E-08
f8	best	9.85E-13	0.00E+00	1.42E-04	1.12E+00	1.80E-13	2.56E-20	0.00E+00
	mean	1.61E-05	8.68E-31	6.40E-01	2.88E+00	1.51E-12	7.46E-09	**0.00E+00**
	std	5.60E-05	6.53E-31	8.41E-01	1.66E+00	8.52E-13	2.09E-08	0.00E+00
f9	best	1.09E-30	1.82E-196	2.31E-08	0.00E+00	1.13E-263	9.86E-148	0.00E+00
	mean	1.71E-11	3.14E-194	5.77E-07	4.44E-183	4.26E-235	1.86E-110	**0.00E+00**
	std	5.80E-11	0.00E+00	8.31E-07	0.00E+00	0.00E+00	8.31E-110	0.00E+00
f10	best	3.93E-08	2.22E-16	5.54E-13	2.22E-16	0.00E+00	0.00E+00	0.00E+00
	mean	8.00E-05	2.22E-16	7.33E-13	3.18E-16	**0.00E+00**	2.00E-16	**0.00E+00**
	std	1.59E-04	0.00E+00	1.18E-13	1.12E-16	0.00E+00	9.93E-17	0.00E+00
f11	best	0.00E+00	3.95E-02	0.00E+00	0.00E+00	0.00E+00	1.23E-02	0.00E+00
	mean	1.16E-04	5.50E-02	5.05E-03	3.61E-02	**0.00E+00**	1.28E-01	**0.00E+00**
	std	2.88E-04	1.27E-02	6.80E-03	5.79E-02	0.00E+00	1.31E-01	0.00E+00
f12	best	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	mean	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
	std	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
f13	best	5.76E-13	1.92E-10	9.95E-01	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	mean	3.57E-05	1.04E+00	3.13E+00	1.94E-01	**0.00E+00**	1.14E+00	**0.00E+00**
	std	1.32E-04	1.03E+00	1.42E+00	8.81E-01	0.00E+00	9.29E-01	0.00E+00
f14	best	3.54E-11	0.00E+00	1.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	mean	1.61E-05	2.00E-01	4.36E+00	6.59E-01	**0.00E+00**	2.05E+00	**0.00E+00**
	std	4.24E-05	4.47E-01	2.04E+00	1.56E+00	0.00E+00	1.54E+00	0.00E+00
f15	best	2.36E-09	4.44E-15	5.84E-13	4.44E-15	8.88E-16	4.44E-15	8.88E-16
	mean	1.41E-03	4.44E-15	7.81E-13	5.84E+00	**8.88E-16**	5.15E-15	**8.88E-16**
	std	1.97E-03	0.00E+00	1.17E-13	8.82E+00	0.00E+00	1.46E-15	0.00E+00
f16	best	7.64E-08	7.29E-107	9.06E-14	3.67E-99	2.40E-74	2.77E-13	2.84E-221
	mean	4.92E-05	8.73E-16	1.39E-13	1.07E-05	**1.14E-70**	3.54E-05	1.78E-16
	std	6.16E-05	1.39E-15	2.51E-14	3.46E-05	3.79E-70	5.71E-05	7.62E-32
f17	best	2.56E-06	2.40E-150	5.69E+03	6.38E-196	5.46E-133	5.14E-67	0.00E+00
	mean	1.30E+00	9.29E-149	9.15E+04	2.36E-112	8.46E-126	1.59E-53	**0.00E+00**
	std	3.06E+00	1.31E-148	8.39E+04	1.26E-111	2.20E-125	6.82E-53	0.00E+00
f18	best	9.11E-08	9.99E-02	9.99E-02	7.96E-02	9.99E-02	9.99E-02	9.99E-02
	mean	**8.71E-05**	1.80E-01	1.68E-01	9.92E-02	9.99E-02	1.60E-01	9.99E-02
	std	1.33E-04	4.47E-02	4.41E-02	3.71E-03	7.06E-17	5.03E-02	8.25E-34
f19	best	2.73E-04	0.00E+00	4.82E-11	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	mean	1.80E-02	**0.00E+00**	1.04E-10	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
	std	1.88E-02	0.00E+00	3.57E-11	0.00E+00	0.00E+00	0.00E+00	0.00E+00
f20	best	6.31E-14	4.71E-32	4.20E-13	3.02E-03	4.07E-26	3.17E-21	4.71E-32
	mean	2.18E-06	5.33E-32	7.55E-03	1.08E-02	8.26E-25	4.67E-02	**4.71E-32**
	std	6.63E-06	1.39E-32	3.38E-02	9.23E-03	1.38E-24	1.14E-01	2.75E-94
f21	best	2.73E-23	1.35E-32	1.93E-23	5.36E-01	5.20E-25	1.50E-23	1.35E-32
	mean	7.81E-06	1.82E-32	2.08E-08	9.37E-01	3.32E-23	1.46E-10	**1.35E-32**
	std	2.48E-05	5.11E-33	8.09E-08	3.10E-01	5.07E-23	6.52E-10	8.03E-96
f22	best	3.34E-11	1.47E-31	8.70E-18	5.53E+00	2.64E-20	4.02E-26	1.35E-31
	mean	1.75E-04	1.63E-05	9.59E-08	8.93E+00	2.52E-17	4.49E-01	**1.35E-31**
	std	4.77E-04	3.64E-05	3.85E-07	2.74E+00	5.52E-17	2.01E+00	0.00E+00
f23	best	2.22E-16	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	mean	2.17E-03	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	1.34E-01	**0.00E+00**
	std	7.72E-03	0.00E+00	0.00E+00	0.00E+00	0.00E+00	6.00E-01	0.00E+00
f24	best	3.55E-15	3.05E-10	1.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	mean	1.09E-05	4.00E-01	3.45E+00	3.69E-01	**0.00E+00**	1.35E+00	**0.00E+00**
	std	2.92E-05	8.94E-01	1.36E+00	1.25E+00	0.00E+00	1.46E+00	0.00E+00

As can be seen from Table 14, in a 10-dimensional problem environment, six state-of-the-art algorithms and ZGBBO all show excellent optimization performance. Especially for the multimodal problem f12, all the seven algorithms successfully jump out of the local optimal state and converge to the global optimal value, with the mean and standard deviation equal to zero. On multimodal problem f23, except ESDA and AOA algorithms, which do not show good performance, the mean value and standard deviation obtained by other algorithms are all 0, and the theoretical optimal solution is successfully searched. In contrast, in the low-dimensional search environment, the MPA showes strong competitiveness, and the mean value obtained on 10 problems is better than other algorithms or the same as ZGBBO, and the mean value is zero on 8 problems, successfully converging to the global optimal value. The mean value of ZGBBO is better than that of other algorithms on 21 test functions, and the mean value of ZGBBO on 15 functions is equal to the theoretical optimal value. Therefore, in the 10-dimensional problem environment, the overall performance of ZGBBO algorithm is better, and other algorithms also show the performance of advanced algorithms.

Table 15 The results of ZGBBO and six state-of-the-art algorithms (30-demensional benchmark functions)

F	Index	ESDA	FDO	AEFA	ChOA	MPA	AOA	ZGBBO
f1	best	1.68E-10	5.98E-92	1.94E-05	8.42E-123	2.58E-108	6.94E-28	0.00E+00
	mean	4.15E-06	2.58E-88	6.98E-02	1.46E-71	3.41E-104	5.37E-26	**0.00E+00**
	std	6.60E-06	5.49E-88	1.23E-01	4.65E-71	1.01E-103	1.35E-25	0.00E+00

f2	best	1.83E-05	2.68E-39	2.12E-12	6.89E-70	1.05E-59	5.30E-17	8.79E-182
	mean	3.63E-03	3.22E-38	2.80E-12	1.17E-44	1.80E-56	9.93E-16	**2.84E-179**
	std	6.60E-03	4.16E-38	3.05E-13	6.18E-44	3.97E-56	1.10E-15	0.00E+00
f3	best	1.12E-08	1.17E-89	3.50E-25	1.71E-96	3.89E-109	1.20E-26	0.00E+00
	mean	1.01E-04	4.13E-85	6.78E-25	5.27E-67	1.24E-102	4.55E-25	**0.00E+00**
	std	4.48E-04	9.20E-85	1.24E-25	2.88E-66	2.94E-102	1.34E-24	0.00E+00
f4	best	7.44E-05	5.01E-04	2.70E-13	1.21E-32	4.30E-41	1.17E-03	5.05E-90
	mean	3.32E-03	3.18E-03	3.36E-13	1.46E-16	1.53E-39	7.27E-03	**3.68E-89**
	std	3.25E-03	2.58E-03	3.56E-14	5.18E-16	2.04E-39	5.31E-03	1.29E-177
f5	best	1.28E-20	5.37E-110	1.71E+03	1.12E-223	1.14E-194	6.21E-42	0.00E+00
	mean	9.55E-05	3.15E-103	3.33E+04	3.84E-123	1.95E-187	2.86E-36	**0.00E+00**
	std	4.69E-04	5.96E-103	2.83E+04	2.10E-122	0.00E+00	9.49E-36	0.00E+00
f6	best	8.71E-07	3.38E-87	1.42E+01	1.17E-102	5.99E-106	6.21E-25	0.00E+00
	mean	1.38E-02	5.59E-83	2.35E+02	1.63E-67	1.90E-101	7.59E-23	**0.00E+00**
	std	3.57E-02	1.24E-82	1.66E+02	8.60E-67	4.44E-101	2.51E-22	0.00E+00
f7	best	1.02E-05	3.69E-01	6.56E-04	1.02E-05	8.43E-05	7.35E-04	1.65E-04
	mean	4.89E-04	6.54E-01	3.05E-03	**8.91E-05**	2.46E-04	2.22E-03	6.53E-04
	std	3.62E-04	2.56E-01	1.95E-03	1.08E-04	1.28E-04	1.27E-03	6.33E-08
f8	best	5.07E-11	3.00E-30	7.37E+00	1.08E+02	3.78E-09	1.58E-03	0.00E+00
	mean	1.25E-04	9.80E-30	3.04E+01	1.43E+02	3.25E-05	7.09E-01	**5.20E-30**
	std	2.58E-04	6.37E-30	1.70E+01	2.20E+01	1.05E-04	1.72E+00	4.75E-59
f9	best	2.63E-22	8.41E-164	5.59E-10	2.71E-222	2.95E-259	6.27E-89	0.00E+00
	mean	3.38E-10	9.15E-160	2.68E-08	9.72E-163	5.79E-234	4.11E-77	**0.00E+00**
	std	1.27E-09	0.00E+00	5.54E-08	0.00E+00	0.00E+00	1.11E-76	0.00E+00
f10	best	2.09E-07	4.44E-16	1.16E-12	4.44E-16	0.00E+00	1.11E-15	0.00E+00
	mean	1.79E-04	6.66E-16	1.39E-12	6.00E-16	**0.00E+00**	2.20E-15	**0.00E+00**
	std	2.13E-04	1.57E-16	1.71E-13	1.03E-16	0.00E+00	8.21E-16	0.00E+00
f11	best	9.88E-10	0.00E+00	6.91E-05	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	mean	6.56E-04	3.53E-03	6.78E-01	9.37E-04	**0.00E+00**	9.68E-03	**0.00E+00**
	std	2.74E-03	4.89E-03	4.47E-01	3.64E-03	0.00E+00	8.27E-03	0.00E+00
f12	best	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	mean	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	2.50E-01	**0.00E+00**
	std	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	6.39E-01	0.00E+00
f13	best	6.48E-10	4.05E+00	9.95E-01	0.00E+00	0.00E+00	1.78E-15	0.00E+00
	mean	1.64E-04	7.94E+00	4.93E+00	**0.00E+00**	**0.00E+00**	3.65E+00	**0.00E+00**
	std	6.53E-04	2.94E+00	2.05E+00	0.00E+00	0.00E+00	2.37E+00	0.00E+00
f14	best	4.11E-09	2.00E+00	3.00E+00	0.00E+00	0.00E+00	1.00E+00	0.00E+00
	mean	5.12E-05	8.60E+00	6.70E+00	6.16E-01	**0.00E+00**	7.25E+00	**0.00E+00**
	std	1.48E-04	3.72E+00	1.89E+00	1.71E+00	0.00E+00	3.64E+00	0.00E+00
f15	best	1.13E-06	7.99E-15	4.20E-13	2.00E+01	8.88E-16	5.77E-14	4.44E-15
	mean	7.85E-04	1.30E-14	5.12E-13	2.00E+01	**3.14E-15**	2.70E-13	4.44E-15
	std	9.62E-04	3.18E-15	5.76E-14	1.42E-03	1.74E-15	3.82E-13	0.00E+00
f16	best	1.63E-07	8.74E-16	1.71E-03	1.43E-57	3.31E-60	5.77E-04	5.13E-181
	mean	2.36E-04	4.18E-15	9.66E-03	1.81E-07	5.04E-57	3.21E-03	**2.04E-101**
	std	2.83E-04	2.81E-15	5.03E-03	9.92E-07	1.15E-56	4.24E-03	3.46E-201
f17	best	6.48E-05	5.80E-86	1.08E+05	2.58E-93	1.88E-104	6.18E-24	0.00E+00
	mean	1.53E+01	3.72E-82	3.79E+05	1.66E-67	2.58E-99	4.70E-22	**3.35E-305**
	std	4.51E+01	5.34E-82	1.79E+05	8.84E-67	9.39E-99	8.37E-22	0.00E+00
f18	best	4.06E-06	4.00E-01	2.00E-01	9.99E-02	9.99E-02	3.00E-01	9.99E-02
	mean	**3.23E-04**	4.80E-01	3.08E-01	9.99E-02	9.99E-02	3.85E-01	1.10E-01
	std	5.63E-04	1.30E-01	5.20E-02	2.44E-09	7.06E-17	4.89E-02	1.00E-03
f19	best	4.57E-03	0.00E+00	1.22E-10	0.00E+00	0.00E+00	1.42E-14	0.00E+00
	mean	5.36E-02	**0.00E+00**	1.96E-10	**0.00E+00**	**0.00E+00**	3.37E-06	**0.00E+00**
	std	5.44E-02	0.00E+00	4.33E-11	0.00E+00	0.00E+00	7.72E-06	0.00E+00
f20	best	2.27E-11	1.57E-32	3.61E-04	6.72E-02	6.18E-13	1.76E-07	1.57E-32
	mean	9.43E-07	5.17E-06	4.97E-02	1.30E-01	3.29E-12	1.56E-02	**1.94E-32**
	std	3.19E-06	9.76E-06	8.12E-02	9.79E-02	1.62E-12	3.80E-02	4.09E-65
f21	best	1.07E-11	1.60E-32	3.67E-05	2.47E+00	1.38E-11	2.44E-06	1.35E-32
	mean	1.44E-05	5.34E-19	4.79E-01	3.60E+00	3.51E-11	1.66E-03	**1.35E-32**
	std	3.65E-05	1.69E-18	6.61E-01	1.29E+00	2.01E-11	4.02E-03	8.32E-96
f22	best	7.60E-09	3.83E+01	4.80E-06	2.62E+01	2.45E-10	4.15E+00	1.35E-31
	mean	3.38E-04	7.23E+02	5.85E-04	1.40E+05	1.54E+00	1.57E+02	**1.35E-31**
	std	5.98E-04	6.77E+02	1.37E-03	1.25E+05	3.97E+00	2.44E+02	0.00E+00
f23	best	2.94E-08	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	mean	2.97E-03	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	2.55E-16	**0.00E+00**
	std	8.73E-03	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.47E-16	0.00E+00
f24	best	4.45E-10	5.00E+00	0.00E+00	0.00E+00	0.00E+00	3.38E-14	0.00E+00
	mean	4.82E-05	9.42E+00	**0.00E+00**	3.63E-01	**0.00E+00**	8.40E+00	**0.00E+00**
	std	8.66E-05	3.54E+00	0.00E+00	1.51E+00	0.00E+00	4.33E+00	0.00E+00

As can be seen from Table 15, in the 30-dimensional problem environment, ZGBBO algorithm maintains excellent search performance, and its mean values on 21 problems are optimal among seven comparison algorithms. Compared with the 10-dimensional problem environment, the convergence performance of ZGBBO remains stable, while the searching ability of the other six state-of-the-art algorithms decreases to varying degrees. For example, the MPA obtains better results on nine test problems than the other comparison algorithms, which is less superior than the calculation results in the 10-dimensional problem environment. But as advanced algorithms, they still keep high convergence accuracy on some problems. For instance, for multimodal problems f12, f14, and f23, most of the six excellent algorithms give the same results as 10 dimensions, with a mean value of zero. ChOA still has the smallest mean value on f7, and ESDA still has the highest convergence accuracy on f18, which is at least 4 exponential levels higher than other algorithms. In contrast, the convergence accuracy of ZGBBO does not decrease obviously, which is basically the same as the calculation result in the 10-dimensional problem environment. Therefore, in the 30-dimensional search environment, ZGBBO still maintains superior search performance and has better robustness.

Table 16 The results of ZGBBO and six state-of-the-art algorithms (50-demensional benchmark functions)

F	Index	ESDA	FDO	AEFA	ChOA	MPA	AOA	ZGBBO
f1	best	2.59E-08	3.13E-40	3.94E-02	1.76E-70	4.30E-101	2.04E-19	3.16E-298
	mean	2.24E-05	3.68E-37	5.41E-01	1.06E-52	6.79E-97	5.62E-18	**8.65E-295**
	std	5.98E-05	7.38E-37	4.47E-01	3.21E-52	1.81E-96	7.11E-18	0.00E+00
f2	best	3.22E-05	1.22E-19	3.26E-12	1.63E-42	5.29E-58	1.15E-11	6.52E-171
	mean	4.64E-03	2.00E-18	3.81E-12	6.25E-34	2.16E-53	2.98E-11	**1.06E-169**
	std	6.55E-03	3.36E-18	3.20E-13	1.52E-33	7.24E-53	1.47E-11	0.00E+00
f3	best	6.35E-10	4.12E-39	7.37E-25	3.65E-67	1.42E-101	7.30E-19	2.23E-299
	mean	1.40E-04	2.37E-37	1.07E-24	1.64E-54	1.25E-96	1.43E-17	**1.19E-295**
	std	3.77E-04	3.25E-37	2.52E-25	4.82E-54	2.85E-96	2.80E-17	0.00E+00
f4	best	9.69E-05	7.04E-01	2.77E-13	1.05E-14	5.01E-37	8.83E-02	3.69E-79
	mean	3.32E-03	2.21E+00	3.98E-13	1.48E-12	6.76E-36	2.62E-01	**6.19E-78**
	std	3.73E-03	1.12E+00	4.52E-14	2.83E-12	6.69E-36	9.62E-02	4.90E-155
f5	best	2.63E-13	6.59E-49	1.44E+05	1.81E-138	8.22E-180	8.40E-24	0.00E+00
	mean	4.26E-03	1.64E-40	1.10E+06	1.89E-90	1.12E-171	2.92E-21	**0.00E+00**
	std	2.27E-02	5.03E-40	8.21E+05	6.03E-90	0.00E+00	6.80E-21	0.00E+00
f6	best	1.11E-09	7.07E-36	7.92E+02	5.84E-65	2.86E-97	2.65E-16	3.03E-294
	mean	9.23E-02	1.60E-33	1.80E+03	3.20E-49	5.91E-94	5.35E-15	**3.91E-291**
	std	2.62E-01	2.49E-33	5.50E+02	7.01E-49	1.48E-93	6.27E-15	0.00E+00
f7	best	2.09E-05	9.43E-02	1.40E-03	9.09E-06	7.85E-05	1.59E-03	2.16E-04
	mean	5.36E-04	1.36E-01	5.51E-03	**7.73E-05**	3.09E-04	3.83E-03	6.33E-04
	std	4.11E-04	3.49E-02	3.12E-03	5.33E-05	1.66E-04	1.67E-03	9.66E-08
f8	best	5.00E-07	1.47E-17	2.80E+02	4.90E+01	8.88E-07	6.99E-02	6.09E-18
	mean	4.70E-04	4.45E-12	3.60E+02	5.51E+02	1.01E+00	8.15E+00	**7.28E-16**
	std	1.29E-03	1.41E-11	4.96E+01	1.47E+02	1.49E+00	9.21E+00	6.34E-30
f9	best	8.50E-20	6.94E-132	1.14E-11	0.00E+00	4.13E-251	2.15E-72	0.00E+00
	mean	2.06E-10	8.78E-127	8.66E-09	5.79E-171	2.83E-231	2.23E-58	**0.00E+00**
	std	5.83E-10	2.67E-126	1.39E-08	0.00E+00	0.00E+00	9.97E-58	0.00E+00
f10	best	5.98E-07	8.88E-16	1.57E-12	1.14E-16	0.00E+00	3.67E-13	0.00E+00
	mean	2.52E-04	1.24E-15	2.01E-12	7.26E-16	**0.00E+00**	1.67E-12	1.18E-16
	std	2.35E-04	2.15E-16	1.98E-13	2.00E-16	0.00E+00	1.10E-12	1.27E-32
f11	best	8.85E-08	0.00E+00	1.30E+00	0.00E+00	0.00E+00	4.44E-16	0.00E+00
	mean	3.16E-04	5.42E-03	1.85E+00	**0.00E+00**	**0.00E+00**	9.90E-03	**0.00E+00**
	std	8.26E-04	6.12E-03	4.00E-01	0.00E+00	0.00E+00	9.07E-03	0.00E+00
f12	best	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	mean	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	1.15E+00	**0.00E+00**
	std	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.39E+00	0.00E+00
f13	best	6.85E-09	1.06E+01	0.00E+00	0.00E+00	0.00E+00	2.13E-14	0.00E+00
	mean	7.66E-05	2.28E+01	4.33E+00	**0.00E+00**	**0.00E+00**	1.91E+00	**0.00E+00**
	std	2.52E-04	1.05E+01	2.22E+00	0.00E+00	0.00E+00	1.47E+00	0.00E+00
f14	best	2.87E-09	1.60E+01	3.00E+00	0.00E+00	0.00E+00	1.00E+00	0.00E+00
	mean	2.63E-05	2.30E+01	6.45E+00	1.80E-01	**0.00E+00**	7.40E+00	**0.00E+00**
	std	4.37E-05	5.41E+00	2.21E+00	5.42E-01	0.00E+00	3.95E+00	0.00E+00
f15	best	4.79E-05	7.19E-14	4.13E-13	6.61E-04	8.88E-16	1.47E-10	4.44E-15
	mean	1.74E-03	1.79E-13	4.98E-13	1.86E+01	4.52E-15	1.45E-09	**4.44E-15**
	std	2.25E-03	1.92E-13	4.64E-14	5.15E+00	6.49E-16	2.83E-09	0.00E+00
f16	best	6.00E-06	7.20E-12	4.64E-02	3.88E-38	2.81E-58	2.76E-03	9.12E-172
	mean	2.81E-04	2.37E-10	5.17E-02	1.46E-25	2.04E-54	4.51E-03	**1.62E-124**
	std	2.56E-04	2.79E-10	4.62E-03	4.04E-25	6.40E-54	1.38E-03	3.21E-247
f17	best	2.96E-04	4.05E-37	2.18E+05	6.66E-69	4.21E-95	3.41E-15	6.74E-294
	mean	1.03E+01	1.68E-33	6.16E+05	7.60E-52	2.74E-92	5.17E-14	**6.93E-290**
	std	2.20E+01	3.27E-33	3.09E+05	2.01E-51	7.40E-92	7.53E-14	0.00E+00
f18	best	5.54E-06	5.00E-01	4.13E-01	6.62E-04	9.99E-02	5.00E-01	9.99E-02
	mean	**4.51E-03**	7.00E-01	5.81E-01	9.35E-02	1.07E-01	6.00E-01	1.33E-01
	std	1.81E-02	1.05E-01	1.10E-01	2.57E-02	2.54E-02	9.73E-02	2.30E-03
f19	best	1.25E-02	0.00E+00	1.59E-10	0.00E+00	0.00E+00	3.62E-05	0.00E+00
	mean	1.64E-01	1.28E-14	3.01E-10	2.59E-16	**0.00E+00**	5.01E-04	**0.00E+00**
	std	2.08E-01	1.24E-14	6.53E-11	8.28E-16	0.00E+00	1.27E-03	0.00E+00
f20	best	1.38E-10	2.41E-20	2.10E-02	1.14E-01	2.15E-11	2.81E-06	7.71E-22
	mean	5.88E-07	1.82E-11	8.72E-02	2.86E-01	7.20E-11	3.17E-02	**1.07E-14**
	std	1.48E-06	5.77E-11	5.51E-02	1.26E-01	3.25E-11	3.76E-02	3.42E-27
f21	best	3.21E-12	3.14E-18	2.55E+00	2.39E+00	3.36E-10	4.44E-05	9.02E-21
	mean	6.54E-06	3.42E-07	5.05E+00	6.63E+00	**1.50E-09**	4.87E-02	3.30E-03
	std	1.12E-05	1.08E-06	1.65E+00	2.58E+00	7.22E-10	7.42E-02	3.26E-04

	best	1.46E-08	2.12E+03	1.03E-03	4.80E+01	3.54E+01	3.74E+01	4.67E-17
f22	mean	**6.61E-04**	4.90E+03	2.21E-01	3.88E+05	4.13E+01	5.60E+02	4.59E-01
	std	2.66E-03	2.48E+03	3.73E-01	1.84E+05	2.35E+00	3.49E+02	5.15E-01
	best	7.57E-09	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.67E-15	0.00E+00
f23	mean	8.61E-03	**0.00E+00**	3.69E-02	**0.00E+00**	**0.00E+00**	6.09E-15	**0.00E+00**
	std	1.72E-02	0.00E+00	1.15E-01	0.00E+00	0.00E+00	8.62E-15	0.00E+00
	best	2.04E-08	1.54E+01	1.00E+00	0.00E+00	0.00E+00	2.52E-09	0.00E+00
f24	mean	5.00E-05	2.53E+01	6.05E+00	6.87E-01	**0.00E+00**	7.05E+00	**0.00E+00**
	std	1.40E-04	1.12E+01	2.42E+00	1.64E+00	0.00E+00	4.66E+00	0.00E+00

It can be seen from Table 16 that in the 50-dimensional search environment, ZGBBO still has the best search performance among seven comparison algorithms, and the mean value obtained on 18 test functions is the smallest, and on nine questions is zero. This shows that in 50 searches, the ZGBBO converges to the theoretical optimal value every time. The ESDA keeps the best convergence precision on the function f18, while the competitiveness of MPA algorithm decreases obviously, which only the mean and standard deviation obtained on the function f21 are better than that of ZGBBO. Tables 14-16 show that with the increase of problem dimension, the solving ability of various algorithms decreases to varying degrees. However, the algorithm performance of ZGBBO is not affected by the problem dimension, and it can still maintain outstanding search performance in high dimensional problem space. In order to fully compare the performance of ZGBBO with the six state-of-the art algorithms, Friedman test is executed on the seven algorithms according to the optimization results of the algorithms in Table 16 in a 50-dimensional search space. The specific results are shown in Table 17 and Fig. 9.

Table 17 The Friedman test of BBO and six state-of-the-art algorithms according to the Table 16

	ESDA	FDO	AEFA	ChOA	MPA	AOA	ZGBBO
f1	6	4	7	3	2	5	1
f2	7	4	5	3	2	6	1
f3	7	4	5	3	2	6	1
f4	5	7	3	4	2	6	1
f5	6	4	7	3	2	5	1
f6	6	4	7	3	2	5	1
f7	3	7	6	1	2	5	4
f8	3	2	6	7	4	5	1
f9	6	4	7	3	2	5	1
f10	7	4	6	3	1	5	2
f11	4	5	7	1	1	6	1
f12	1	1	1	1	1	7	1
f13	4	7	6	1	1	5	1
f14	3	7	5	4	1	6	1
f15	6	3	4	7	2	5	1
f16	5	4	7	3	2	6	1
f17	6	4	7	3	2	5	1
f18	1	7	5	2	3	6	4
f19	7	4	5	3	1	6	1
f20	4	2	6	7	3	5	3
f21	3	2	6	7	1	5	4
f22	1	6	2	7	4	5	3
f23	6	1	7	1	1	5	1
f24	3	7	5	4	1	6	1
Mean Rank	4.58	4.33	5.50	3.50	1.88	5.46	1.58
Final Rank	5	4	7	3	2	6	1

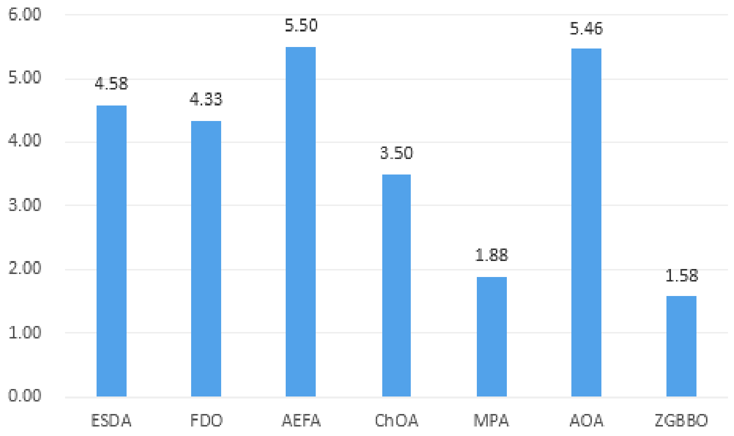
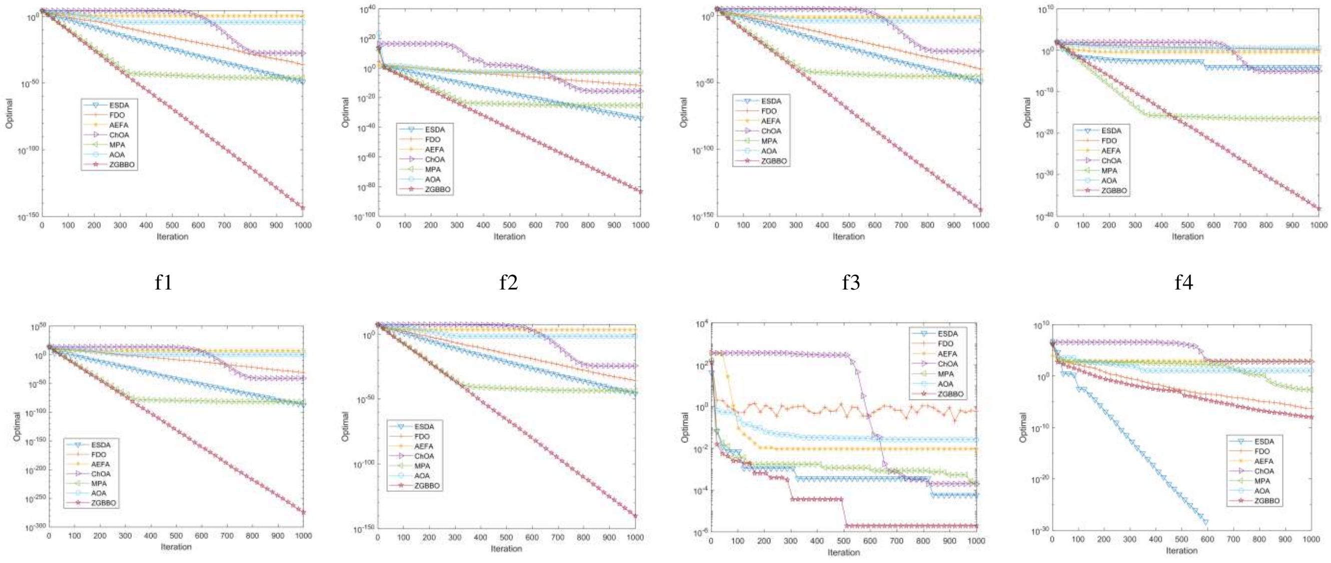


Fig. 9 Average ranking of ZGBBO and six state-of-the-art algorithms

As can be seen from Table 17, the optimization result of ZGBBO algorithm on 18 benchmark functions is the best among seven evolutionary algorithms, and the success rate of ZGBBO algorithm in six state-of-the-art algorithms reaches 75%. According to the average ranking of six comparison algorithms in Fig. 9, MPA proposed in 2020 is the most competitive among all competitors, ranking only second to ZGBBO on average, and is an excellent algorithm among the new evolutionary algorithms proposed in recent two years. In contrast, the search results of AOA proposed in 2021 on 24 test functions are only better than AEFA proposed in 2019. The rest of the state-of-the-art algorithms show unique high performance and excellent search ability on different text functions, but they also have low convergence accuracy and slow search speed on other text functions, and their comprehensive competitiveness is general. According to Table 17 and Fig. 9, ZGBBO has the best overall performance among the seven evolutionary algorithms and is more competitive than the other six state-of-the-art algorithms. Therefore, ZGBBO is a new evolutionary algorithm with advanced and superiority.

5.4.2 Comparison of convergence rates

In order to fully verify the advancement of ZGBBO, we compare the convergence rate of ZGBBO and six state-of-the-art evolutionary algorithms on 24 different benchmark functions of 50 dimensions in this section. If section 5.4.1 compares the convergence accuracy of the algorithms under the same maximum evaluation times, then this section compares the convergence speed of the algorithms under the same iteration times. The specific operation is the same as the section 5.2.3, the maximum iteration times *T*=1000 and the population size *NP*=5*D* are set, and each algorithm is independently run for 50 times. The optimal running result and the calculation results of each generation are recorded, and the optimal convergence image of seven comparison algorithms on the high-dimensional test function is drawn, as shown in Fig. 10.



f1

f2

f3

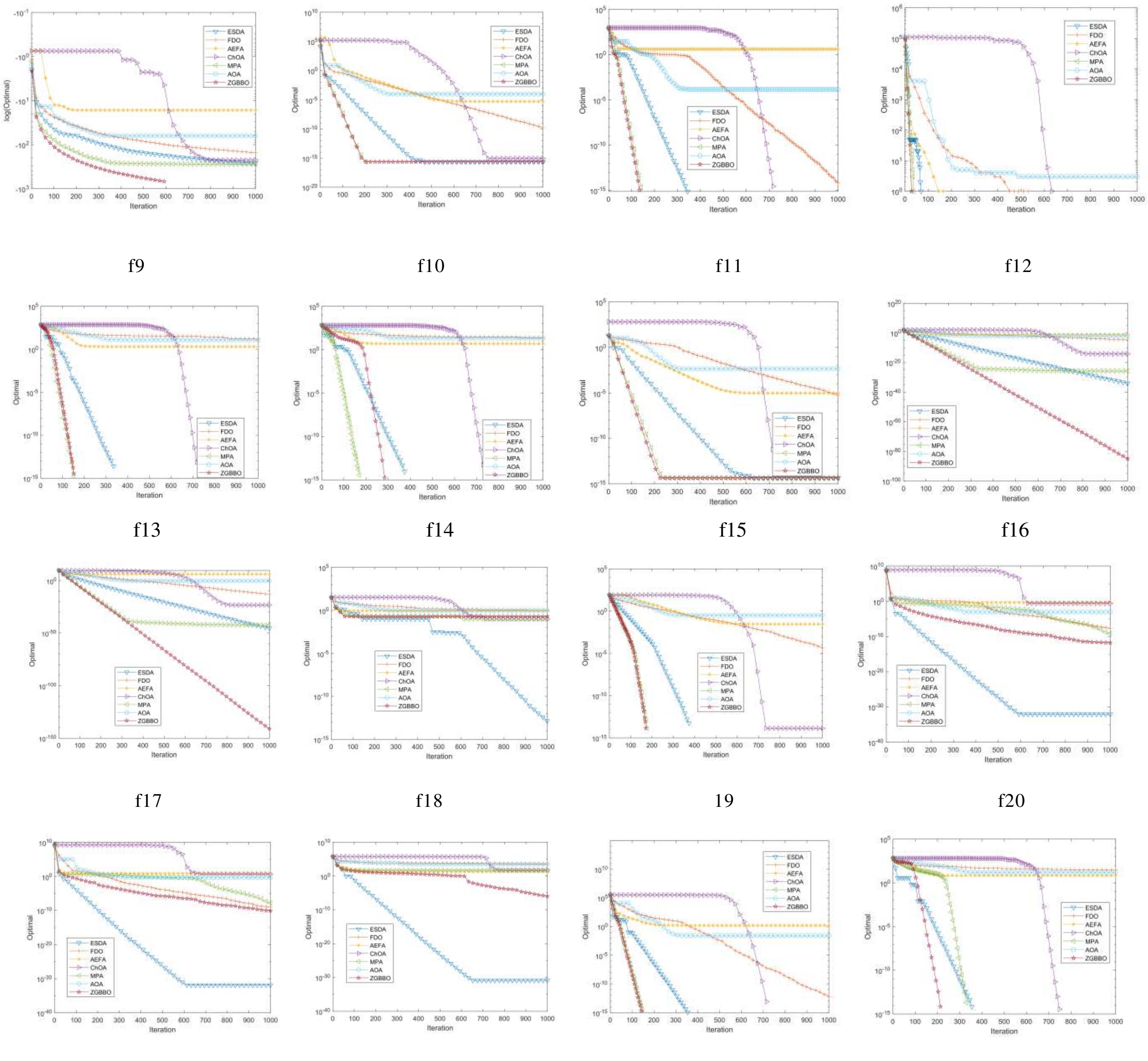
f4

f5

f6

f7

f8



f9 f10 f11 f12

f13 f14 f15 f16

f17 f18 19 f20

f21 f22 f23 f24

Fig. 10 Convergence of seven evolutionary algorithms on different test functions

It can be observed from Fig. 10 that under the same number of evolutions, ZGBBO has a faster convergence speed than the other 6 state-of-the-art algorithms on 75% of the test functions. Especially for the single-peak functions f1-f10, the convergence curve of ZGBBO decreases rapidly and is obviously better than other algorithms. Although the convergence speed of ZGBBO on f8 is not the fastest among the seven algorithms, it is second only to the electrostatic discharge algorithm. For multimodal problems, the ZGBBO algorithm combined with differential evolution shows outstanding performance among six new evolutionary algorithms. For example, on the multi-modal functions f11-f13, f15-f17, f19, f23 and f24, ZGBBO does not fall into the local optimal solution like other evolutionary algorithms, but rapidly converges to the global optimal solution, and the convergence curve decreases significantly. In contrast, the electrostatic discharge algorithm proposed in 2019 and the marine predator algorithm proposed in 2020 also show strong competitiveness in convergence speed. For example, on f11-13, f15, f19 and f23, the convergence curve of MPA almost coincides with the convergence curve of ZGBBO. On multi-modal function f14, the convergence curve of MPA decreases faster than that of ZGBBO, while on f20-f22, the convergence rate of ESDA algorithm is the most ideal among the seven comparison algorithms. Even so, the convergence speed of ZGBBO on multi-modal functions f14 and f20-f22 is only second to MPA or ESDA, and better than other state-of-the art algorithms. In addition, ESDA algorithm has the best convergence speed on 20.8% of the test questions, while ZGBBO has the fastest convergence speed on 75% of the test questions. Therefore, the convergence of ZGBBO is still more competitive than the other six state-of-the-art evolutionary algorithms, and the algorithm has better performance on the whole. So ZGBBO is a ideal algorithm to choose and use.

6 ZGBBO complexity analysis

The evaluation of an algorithm includes three aspects: optimization performance, convergence speed and complexity. Previously, we have verified that ZGBBO has excellent optimization performance and search ability. Now, the complexity of ZGBBO is analyzed to fully prove the effectiveness and advancement of ZGBBO algorithm. In the same development environment, the number of function evaluations and the calculation design of the algorithm jointly determine the running time of the algorithm. However, in the experimental settings of this paper, all the algorithms set the same maximum evaluation times for each benchmark function, so the reason for ZGBBO's fast convergence speed and short running time is not caused by the use of more function evaluation times, but that this paper fully reduces the computational complexity in the search process of ZGBBO. We will analyze and discuss it in detail in this chapter.

6.1 Time consumption

In the experiment of the Chapter 5, we can find ZGBBO algorithm's convergence speed is quicker than the other algorithms. But the algorithm convergence speed running time is short, doesn't mean algorithm complexity is low. In real life there are many problems need to be within a certain amount of time we find a solution. Therefore, sacrifice a lot of time to get the solution is not desirable. In order to verify that ZGBBO is also an acceptable algorithm in terms of time consumption, we compare and analyze the running time of ZGBBO and original BBO. We measure the average CPU running time of each iteration when BBO and ZGBBO are solving 24 benchmark problems, so as to compare the results of time sacrifice of the algorithm, as shown in Fig. 11.

As can be seen from Fig. 11, although several improvement strategies are added to ZGBBO, the time consumption of the algorithm does not increase. Moreover, on some functions, such as f19, ZGBBO has a shorter average running time than BBO. This is because when ZGBBO adds the improved strategy, it also deletes the mutation operator and other calculation steps of the original BBO, so as to balance the running time of the algorithm. Combined with the above experimental results, it can be found that ZGBBO can obtain higher precision problem solutions in the same or shorter time as the original BBO.

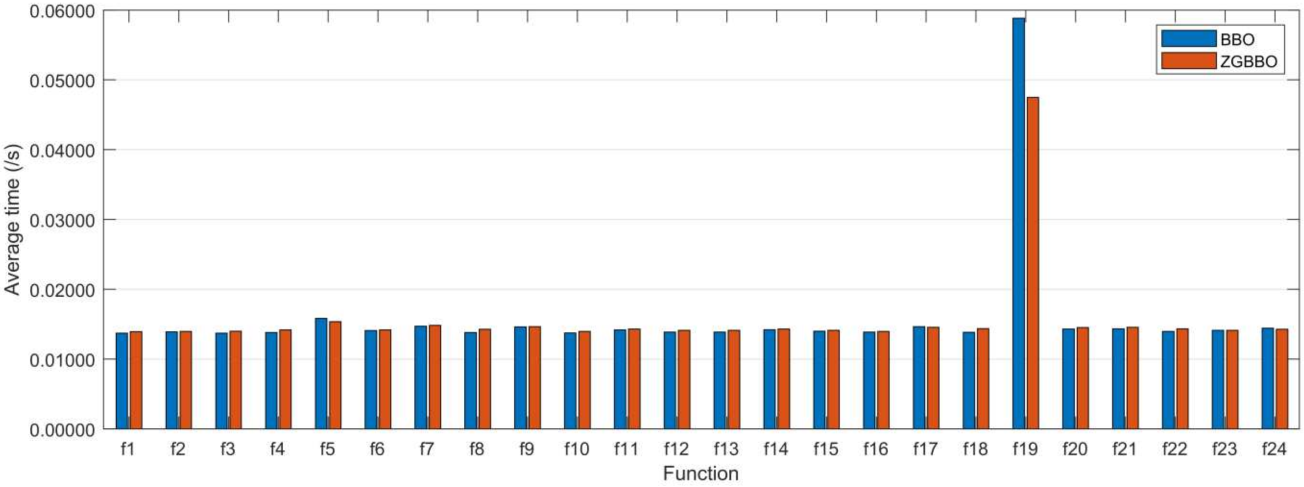


Fig. 11 Average runtime comparisons among BBO and ZGBBO on the benchmark functions

6.2 calculated amount

To verify the comparison results in Fig. 11, we will analyze the calculation steps of ZGBBO and original BBO in depth. In order to facilitate intuitive comparison and understand the differences between two algorithms, we transform the calculation steps of BBO and ZGBBO into flow charts, as shown in Fig. 12. Based on the observation of Fig. 12, we carry out the following analysis and discussion.

First of all, in each iteration of the original BBO algorithm, the immigration rate and emigration rate of each habitat in the population need to be recalculated, so the total calculation times are $2 \cdot NP \cdot T$. But the immigration rates and emigration rates are based on rankings. In other words, the immigration rates and emigration rates are only related to the ranking of habitats. According to Eqs. (2) and (3), as long as the ranking of individuals is determined, the immigration rate and emigration rate of corresponding individual can be determined. Therefore, in the iteration process of ZGBBO, no matter how many times the total iteration is, the immigration rates and emigration rates of all habitats in the population are calculated only once. In addition, ZGBBO adopts the example learning method to select habitats for migration, which does not need to calculate the emigration rate of each habitat, further reducing the calculation amount, and the total calculation times is *NP*. Therefore, compared with BBO, ZGBBO greatly reduces the operational complexity in the calculation of the immigration rate and the emigration rate, saving at least 2 • (*NP*-1) • *T* times calculation.

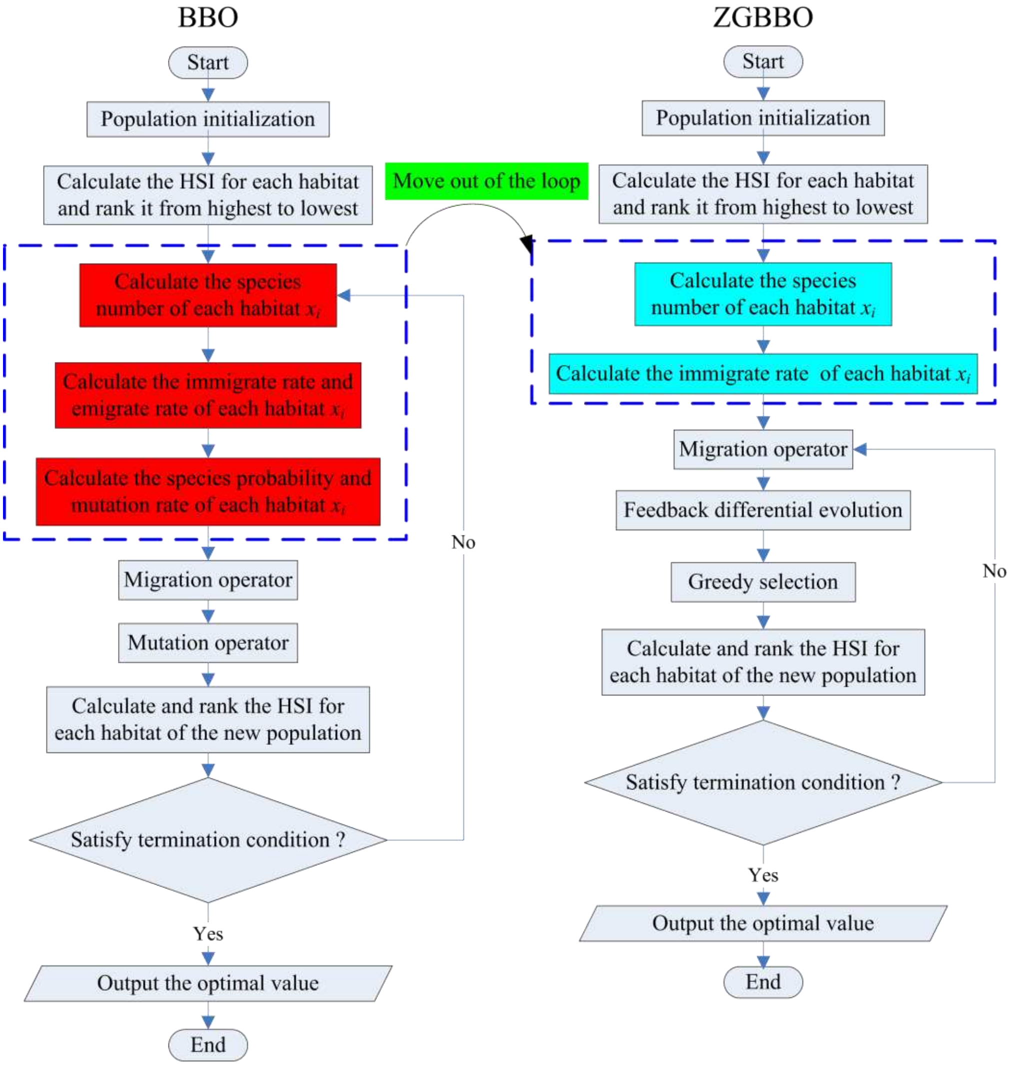

BBO
Start
Population initialization
Calculate the HSI for each habitat and rank it from highest to lowest
Calculate the species number of each habitat x_i
Calculate the immigrate rate and emigrate rate of each habitat x_i
Calculate the species probability and mutation rate of each habitat x_i
Migration operator
Mutation operator
Calculate and rank the HSI for each habitat of the new population
Satisfy termination condition ?
No
Yes
Output the optimal value
End
Move out of the loop
ZGBBO
Start
Population initialization
Calculate the HSI for each habitat and rank it from highest to lowest
Calculate the species number of each habitat x_i
Calculate the immigrate rate of each habitat x_i
Migration operator
Feedback differential evolution
Greedy selection
Calculate and rank the HSI for each habitat of the new population
Satisfy termination condition ?
No
Yes
Output the optimal value
End

Fig. 12 Flowchart of BBO and ZGBBO

Secondly, the mutation rate of each habitat only needs to be calculated once. According to Eqs. (5) and (6), as long as the immigration rate and emigration rate are determined, the mutation rate of a habitat can be obtained. Therefore, habitat mutation rate is also based on habitat suitability ranking and does not need to be calculated repeatedly in each iteration. However, the original BBO does not avoid repeated calculation, but calculates the mutation rate of each habitat in every generation. The species probability of the habitat should be calculated first to calculate the mutation rate, that is, the calculation times of each generation is 2 • *NP*, so the total calculation times of BBO mutation operator is at least 2 • *NP* • *T*. By contrast, ZGBBO directly deletes the mutation operator in the search process and uses the feedback differential evolution mechanism to replace it. Although the standard deviation of population suitability needs to be calculated, the amount of calculation is far less than the species probability. ZGBBO calculates 0.8 time's standard deviation for the whole population on average in each iteration, that is, the total calculation times is 0.8 • *T*. Therefore, ZGBBO greatly reduces the computational complexity and saves at least 2 • *NP* • (*T*-0.8) calculations in the mutation step. In addition, ZGBBO has one more judgment step than BBO in the migration process, but it does not introduce additional loops or add additional calculations. BBO used roulette to select the habitat to be emigrated, and the calculation times of each generation is *NP*, so the total calculation times is *NP* • *T*. Then, ZGBBO adopts the example learning method to select the emigration habitat. According to Eq. (7), each individual only needs to calculate once to select the habitat for migration. Therefore, the total calculation times of ZGBBO in the migration process is also *NP* • *T* times, and the calculation amount does not increase. Finally, ZGBBO adds a opposition-based learning strategy, which generates opposite individuals for the half of individuals with poor fitness in each iteration. The algorithm generates an opposite individuals would increase one calculation, and the calculation times of each generation would increase to *NP*/2, so the total calculation times is $NP \cdot T/2$. Although ZGBBO increases $NP \cdot T/2$ computation in the opposition-based learning, this has been compensated in the other operators mentioned above.

To sum up, ZGBBO reduces the computational complexity of the algorithm from several parts and achieves the purpose of reducing the computational amount. Although the amount of calculation is increased in the process of generating opposite individuals, it is fully compensated in other parts. Compared with the original BBO algorithm, the calculation times saved by ZGBBO algorithm is at least:

$2 \cdot (NP-1) \cdot T+2 \cdot NP \cdot (T-0.8) - NP \cdot T/2=(3.5T-1.6) \cdot NP-2T$.

7 Conclusions and future research

In this paper, we analyze the performance deficiencies of the standard BBO algorithm, and propose a new biogeography-based optimization named ZGBBO. The framework of ZGBBO mainly includes three operators: selection, migration and feedback differential evolution. The selection operator and migration operator are used to improve the convergence accuracy and speed of the algorithm, so as to enhance the optimization efficiency. The feedback differential evolution is used to help the algorithm escape the local optimal solution, so as to enhance the optimization ability of multi-modal problems. To reduce the computational complexity, the mutation operator of BBO is deleted. Meanwhile, we creatively establish a population-based sequence convergence model to prove the convergence of the algorithm. In order to verify the effectiveness and necessity of three improved operators, simulation experiments are carried out on ZGBBO and its three variants. Experimental results on 24 benchmark functions show that the three improved strategies are indispensable. In addition, ZGBBO is compared with seven improved BBO variants and six state-of-the-art evolutionary algorithms. On the whole, ZGBBO has the best overall performance. Finally, the algorithm complexity of ZGBBO is analyzed. By comparing with the original BBO, the effectiveness of the proposed algorithm is fully verified.

With the rapid development of modern society, practical problems in real life have higher and higher requirements for algorithms. Although the ZGBBO algorithm presented in this paper shows good results on high-dimensional test functions, it has not been applied to practical problems. An excellent algorithm need to serve human life. Therefore, the next step will be to further study the defects of ZGBBO algorithm, improve its optimization ability, and fully reduce the running time. In the future, ZGBBO will be applied to solve more complex practical problems, such as engineering optimization, wireless sensor network coverage, environmental monitoring, etc.

Authorship contribution statement

Ziyu Zhang: Investigation, Methodology, Experiment, Writing: original draft. **Yuelin Gao**: Supervision, Funding acquisition. **Jiahang Li**: Project administration, Writing: review and editing. **Wenlu Zuo**: Data curation, Writing: review and editing.

Acknowledgements

This work is proudly supported in part by the National Natural Science Foundation of China under Grant (11961001), the Construction Project of First-class Subjects in Ningxia Higher Education (NXYLXK2017B09) and the Major Proprietary Funded Project of North Minzu University (ZDZX201901).

Declaration of interests

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 paper. All authors guarantee that the paper 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 author himself will be responsible for any economic losses and negative social impacts caused by bad behavior.

Reference

Awad N, Ali M, Liang J, et al. (2017) Problem definitions and evaluation criteria for the CEC 2017 special session and competition on singe objective bound constrained real-parameter numerical optimization. Technical report. Singapore: Nanyang Technological University.

Anita, Yadav A. (2019) AEFA: Artificial electric field algorithm for global optimization. Swarm and Evolutionary Computation 48(2019): 93-108. https://doi.org/10.1016/j.swevo.2019.03.013

Ali Reihanian, Mohammad-Reza Feizi-Derakhshi,Hadi S. Aghdasi. (2019) NBBO: A new variant of biogeography-based optimization with a novel framework and a two-phase migration operator. Information Sciences 504(2019): 178-201. https://doi.org/10.1016/j.ins,2019.07.054.

Abdullah J M , Rashid T A. (2019) Fitness Dependent Optimizer: Inspired by the Bee Swarming Reproductive Process. IEEE Access 22(2019): 43473-43486. https://doi.org/10.1109/ACCESS.2019.2907012

Albashish D, Hammouri A I, Braik M, et al. (2020) Binary biogeography-based optimization based SVM-RFE for feature selection. Applied Soft Computing 101: 107026. https://doi.org/10.1016/j.asoc.2020.107026

An Y, X Chen, Li Y, et al. (2020) An improved non-dominated sorting biogeography-based optimization algorithm for the (hybrid) multi-objective flexible job-shop scheduling problem. Applied Soft Computing 99(C):106869. https://doi.org/10.1016/j.asoc.2020.106869

Dorigo M. (1992) Optimization, learning and natural algorithms. Ph. D. Thesis, Politecnico di Milano.

Dasgupta D. (1998) An artificial immune system as a multi-agent decision support system. IEEE International Conference on Systems. IEEE 1998: 3816-3820. https://doi.org/10.1109/ICSMC.1998.726682

Dhiman G, Kumar V. (2019) Seagull optimization algorithm: Theory and its applications for large-scale industrial engineering problems. Knowledge-Based Systems 165(FEB.1): 169-196. https://doi.org/10.1016/j.knosys.2018.11.024

Ergezer M. (2014) Oppositional biogeography-based optimization. Dissertation, Cleveland State University. https://doi.org/10.1109/ICSMC.2009.5346043

Erlich I, Rueda J L, Wildenhues S, et al. (2014) Evaluating the mean-variance mapping optimization on the IEEE-CEC2014 test suite. IEEE Evolutionary Computation: 1625-1632. https://doi.org/10.1109/CEC.2014.6900516

Elsisi M . (2018) Future search algorithm for optimization. Evolutionary Intelligence 12:21–31. https://doi.org/10.1007/s12065-018-0172-2

Feng QX, Liu SY, Zhang JK, Yang GP, Yong LQ. (2017) Improved biogeography-based optimization with random ring topology and Powell's method. Applied Mathematical Modelling 41(2017): 630-649. https://dx.doi.org/10.1016/j.apm.2016.09.020

Faramarzi A, Heidarinejad M, Mirjalili S, et al. (2020) Marine Predators Algorithm: A Nature-inspired Metaheuristic. Expert Systems with Applications 152: 113377. https://doi.org/10.1016/j.eswa.2020.113377

Guo WA, Chen M, Wang L, Mao YF, Wu QD. (2017) A survey of biogeography-based optimization. Neural Computing and Applications 28(8): 1909-1926. https://doi.org/10.1007/s00521-016-2179-x

Ghatte H F. (2021) A Hybrid of Firefly and Biogeography-Based Optimization Algorithms for Optimal Design of Steel Frames. Arabian Journal for Science and Engineering 46(5): 4703-4717. https://doi.org/10.1007/s13369-020-05118-w

Holland J. (1975) Adaptation in natural and artificial systems: an introductory analysis with application to biology. Control & Artificial Intelligence.

Houssem R.E.H Bouchekara. (2019) Electrostatic discharge algorithm: a novel nature-inspired optimisation algorithm and its application to worst-case tolerance analysis of an EMC filter. IET Science, Measurement & Technology 13(4): 491-499. https://doi.org/10.1049/iet-smt.2018.5194

Hashim F A, Hussain K, Houssein E H, et al. (2021) Archimedes optimization algorithm: a new metaheuristic algorithm for solving optimization problems. Applied Intelligence 51(3): 1531-1551. https://doi.org/10.1007/s10489-020-01893-z

J. Kennedy , R. Eberhart. (1995) Particle swarm optimization, in: IEEE International Joint Conference on Neural Networks, Perth, Western Australia pp. 1942–1948.

Joaquín Derrac, Salvador García, C D M, et al. (2011) A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. Swarm & Evolutionary Computation 1(1): 3-18. https://doi.org/10.1016/j.swevo.2011.02.002

Kingman J F C. (1981) Finite Markov Processes and Their Applications. Bulletin of the London Mathematical Society (S1469-2120) 13(3): 250-251. https://doi.org/10.2307/2530590

Kang S L, Zong W G (2005) A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice[J]. Computer Methods in Applied Mechanics & Engineering 194(36-38): 3902-3933. https://doi.org/10.1016/j.cma.2004.09.007

Khishe M, Mosavi M R. (2020) Chimp optimization algorithm. Expert Systems with Applications 149: 113338. https://doi.org/10.1016/j.eswa.2020.113338

Liao T, Stutzle T. (2013) Benchmark results for a simple hybrid algorithm on the CEC 2013 benchmark set for real-parameter optimization. IEEE Evolutionary Computation: 1938-1944.

Loon L W, Antoni W, Ishak D M, et al. (2016) A Biogeography-Based Optimization Algorithm Hybridized with Tabu Search for the Quadratic Assignment Problem. Computational Intelligence and Neuroscience 2016: 5803893. https://doi.org/10.1155/2016/5803893

Liu S, Wang P, Zhang J. (2018) An Improved Biogeography-Based Optimization Algorithm for Blocking Flow Shop Scheduling Problem. Chinese Journal of Electronics 27(002): 351-358. https://doi.org/10.1049/cje.2018.01.007

Ma HP. (2010) An analysis of the equilibrium of migration models for biogeography-based optimization. Information Sciences 180(18): 3444-3464. https://doi.org/10.1016/j.ins.2010.05.035

Ma HP, Simon D, Patrick Siarry, Yang ZL, Fei MR. (2017) Biogeography-Based Optimization: A 10-Year Review. IEEE Trans. Emerging Topics in Comput. Intellig 1(5): 391-407. https://doi.org/10.1109/TETCI.2017.2739124

Mirjalili S, Lewis A. (2016) The whale optimization algorithm. Advances in engineering software 95: 51-67. https://doi.org/10.1016/j.advengsoft.2016.01.008

Mohamed A, Mohamed Y S, El-Gaafary A A M, et al. (2017) Optimal power flow using moth swarm algorithm. Electric Power Systems Research 142(JAN): 190-206. https://doi.org/10.1016/j.epsr.2016.09.025

Mehta K, Pal R. (2018) Modified Biogeography Based Optimization (MBBO). International Journal on Future Revolution in Computer Science & Communication Engineering 4(1): 124-129. https://doi.org/10.13140/RG.2.2.21429.78562

Passino K M. (2006) Biomimicry of Bacterial Foraging for Distributed Optimization and Control. IEEE Control Systems Magazine 22(3), 52-67. https://doi.org/10.1109/MCS.2002.1004010

Pushpa Farswan, Jagdish Chand Bansal. (2019) Fireworks-inspired biogeography-based optimization. Soft Computing 23(16): 7091-7115. https://doi.org/10.1007/s00500-018-3351-2

Storn R, Price K. (1995) Differential Evolution: A Simple and Efficient Adaptive Scheme for Global Optimization Over Continuous Spaces. Journal of Global Optimization 23(1): 1-12.

Simon D. (2008) Biogeography-Based Optimization. IEEE Transactions on Evolutionary Computation 12(6): 702-713. https://doi.org/10.1109/TEVC.2008.919004

Sloss A N, Gustafson S. (2020) 2019 evolutionary algorithms review. Genetic Programming Theory and Practice XVII, 2020: 307. ISBN: 978-3-030-39957-3

Sang X, Liu X, Zhang Z, et al. (2021) Improved Biogeography-Based Optimization Algorithm by Hierarchical Tissue-Like P System with Triggering Ablation Rules. Mathematical Problems in Engineering 2021: 6655614. https://doi.org/10.1155/2021/6655614

Teodorovic D, Lucic P, Markovic G, et al. (2006) Bee Colony Optimization: Principles and Applications. Neural Network Applications in Electrical Engineering, 2006(8):151-156. https://doi.org/10.1109/NEUREL.2006.341200

Tan Y, Shi Y, Tan KC. (2010) Advances in Swarm Intelligence Volume 6145 || Fireworks Algorithm for Optimization. Lecture Notes in Computer Science (Chapter 44): 355-364. https://doi.org/10.1007/978-3-642-13495-1

Xiong GJ, Shi DY. (2018) Hybrid biogeography-based optimization with brain storm optimization for non-convex dynamic economic dispatch with valve-point effects. Energy 157(2018): 424-435. https://doi.org/10.1016/j.energy.2018.05.180

Xue JK, Shen B. (2020) A novel swarm intelligence optimization approach: sparrow search algorithm. Systems Science & Control Engineering 8(1): 22-34. https://doi.org/10.1080/21642583.2019.1708830

Yogesh C K, Hariharan M, Ngadiran R, et al. (2016) A New Hybrid PSO Assisted Biogeography-Based Optimization for Emotion and Stress Recognition from Speech Signal. Expert Systems with Applications 69: 149-158. https://dx.doi.org/10.1016/j.eswa.2016.10.035

Zhang XM, Kang Q, Tu Q, Cheng F, Wang X. (2019a) Efficient and merged biogeography-based optimization algorithm for global optimization problems. Soft Computing 23(12): 4483-4502. https://doi.org/10.1007/s00500-018-3113-1

Zhang X, Wang D, H Chen. (2019b) Improved Biogeography-Based Optimization Algorithm and Its Application to Clustering Optimization and Medical Image Segmentation. IEEE Access 7:28810-28825. https://doi.org/10.1109/ACCESS.2019.2901849

Zhao FA, Qin SA, Zhang YB, et al. (2019) A two-stage differential biogeography-based optimization algorithm and its performance analysis. Expert Systems with Applications 115(2019): 329-345. https://doi.org/10.1016/j.eswa.2018.08.012

Zhao F, Du S , Zhang Y, et al. (2020) Hybrid biogeography-based optimization with enhanced mutation and CMA-ES for global optimization problem. Service Oriented Computing and Applications 14(1): 65-73. https://doi.org/10.1007/s11761-019-00284-8