



Hybrid metaheuristic algorithm using butterfly and flower pollination base on mutualism mechanism for global optimization problems

Zhongmin Wang¹ · Qifang Luo^{1,2} · Yongquan Zhou^{1,2,3}

Received: 22 January 2020 / Accepted: 15 April 2020 / Published online: 8 May 2020
© Springer-Verlag London Ltd., part of Springer Nature 2020

Abstract

The butterfly optimization algorithm (BOA) is a new metaheuristic algorithm that is inspired from food foraging behavior of the butterflies. Because of its simplicity and effectiveness, the algorithm has been proved to be effective in solving global optimization problems and applied to practical problems. However, BOA is prone to local optimality and may lose its diversity, thus suffering losses of premature convergence. In this work, a hybrid metaheuristic algorithm using butterfly and flower pollination base on mutualism mechanism called MBFPA was proposed. Firstly, the flower pollination algorithm has good exploration ability and the hybrid butterfly optimization algorithm and the flower pollination algorithms greatly improve the exploration ability of the algorithm; secondly, the symbiosis organisms search has a strong exploitation capability in the mutualism phase. By introducing the mutualism phase, the algorithm's exploitation capability is effectively increased and the algorithm's convergence speed is accelerated. Finally, the adaptive switching probability is increased to increase the algorithm's balance in exploration and exploitation capabilities. In order to evaluate the effectiveness of the algorithm, in the 49 standard test functions, the proposed algorithm was compared with six basic metaheuristic algorithms and five hybrid metaheuristic algorithms. MBFPA has also been used to solve five classic engineering problems (three-bar truss design problem; multi-plate disc clutch brake design; welded beam design; pressure vessel design problem; and speed reducer design). The results show that the proposed method is feasible and has good application prospect and competitiveness.

Keywords Butterfly optimization algorithm (BOA) · Flower pollination algorithm (FPA) · Mutualism mechanism · Benchmark functions · Engineering design problem · Hybrid metaheuristic

1 Introduction

In real life, optimization is everywhere. Optimization is a process of finding the optimal solution. At present, a large number of optimization methods have been used to deal with optimization problems. Most of the traditional optimization methods rely on gradient information to update the solution, and the position of the initial solution affects the quality of the final solution. Therefore, in many practical engineering

design problems, it is difficult to get a satisfactory solution based on gradient optimization method. Recently, a nature inspired metaheuristic optimization method is becoming more and more popular with the field of optimization and is widely used to solve complex optimization problems of various fields. According to a survey, metaheuristic has solved the optimization problem with sufficient efficiency and reasonable calculation cost compared with the accurate method [1].

The term "metaheuristic" generally refers to the approximate algorithm used for optimization, which is not specifically expressed for a particular problem, some metaheuristics are inspired by natural processes such as evolution, while others are extensions of less complex algorithms such as greedy heuristics and local search [2]. Existing natural metaheuristic algorithms can be simply divided into the following four categories: swarm intelligence based, bio-inspired, physical/chemical inspired and human behaviors. Among the population-based algorithms, some of the more

✉ Yongquan Zhou
yongquanzhou@126.com

¹ College of Artificial Intelligence, Guangxi University for Nationalities, Nanning 530006, Guangxi, China

² Key Laboratories of Guangxi High Schools Complex System and Computational Intelligence, Nanning 530006, Guangxi, China

³ Guangxi Key Laboratories of Hybrid Computation and IC Design Analysis, Nanning 530006, China

classical algorithms are particle swarm optimization (PSO) [3], ant colony optimization (ACO) [4], artificial bee colony (ABC) [5], cuckoo Search (CS) [6], flower pollination algorithm (FPA) [7], firefly algorithm (FA) [8]. In addition, in recent years, some novel and effective swarm algorithms have been proposed, including Grey wolf optimizer (GWO) [9], whale optimization algorithm (WOA) [10], Harris hawks optimization (HHO) [11], symbiotic organisms search (SOS) [12], butterfly optimization algorithm (BOA) [13], etc. Bio-inspired algorithm includes evolutionary algorithms and immune type algorithms: evolutionary algorithms imitate the reproduction, recombination, selection and mutation stages of biological evolution. Some common evolutionary algorithms include genetic algorithms (GA) [14] and differential evolution (DE) [15]; immune type algorithm is to propose optimization algorithm based on immune mechanism and comply with immunological principle. The most representative ones are immune genetic algorithm (IGA) [16], clonal selection algorithm (CSA) [17], negative selection algorithm (NSA) [18], b-cell algorithm (BCA) [19], etc. In the third physical/chemical inspired algorithm, search individuals are updated based on physical phenomena, rules or some chemical reaction rules. For example, simulated annealing (SA) [20], gravitational search algorithm (GSA) [21], lightning search algorithm (LSA) [22], multi-verse optimizer (MVO) [23], electromagnetic field optimization (EFO) [24], equilibrium optimizer (EO) [25], chemical reaction optimization (CRO) [26], artificial chemical reaction optimization algorithm (ACROA) [27], etc. In the last kind of algorithm based on human behavior, it is subdivided in paper [28] social ideologies, sports competitive behavior, social and cultural interaction and condensation. Among them, the representative ones are cultural evolution algorithm (CEA) [29], teaching learning-based optimization (TLBO) [30], imperial list competitive algorithm (ICA) [31], etc.

Butterfly optimization algorithm (BOA) is a metaheuristic algorithm proposed by Arora and Singh. BOA is based on food foraging behavior and information sharing strategy of butterflies [13]. Literature [13] shows that the performance of BOA is better than that of the generally accepted optimization algorithm. Since the BOA algorithm was proposed, in order to obtain better results in the exploitation and exploration capabilities of the algorithm, Arora et al. have made a series of improvements on BOA: literature [32] proposed an improved butterfly optimization algorithm, which adopts the variable sensing modal parameter strategy to improve the convergence speed of the algorithm; literature [33] proposed an improved butterfly optimization algorithm with chaos; the algorithm's exploration capability has been increased; literature [34] proposed a hybrid method BOA/DE by the ensemble of BOA and DE algorithm, which combines the advantages of BOA and DE to

enable the algorithm to achieve between exploration and exploitation balance to produce efficient results; literature [35] proposed the hybrid method BOA/ABC by the ensemble of BOA and ABC algorithm, which is similar to BOA/DE; literature [36] introduced learning automata, which not only keeps the main characteristics of the basic BOA, but also accelerates the global convergence speed and achieves the real global optimization; literature [37] proposed an modified BOA (MBOA), which adds the modification operation to the optimal position, thereby increasing the algorithm exploitation capability. In addition, Singh et al. [38] proposed an adaptive butterfly optimization algorithm, which improved the convergence speed of the algorithm by changing the sensory mode of BOA; Li [39] proposed an improved butterfly optimization algorithm (BOA), by embedding the cross-entropy (CE) method into the BOA, and the results showed that the improved algorithm achieved a good balance in exploration and exploitation; Sharma and Sushmita [40] proposed a butterfly optimization algorithm enhanced with mutualism scheme which improves the exploitation ability of BOA. Sharma et al. [41] proposed an integrated algorithm of butterfly optimization algorithm and symbiotic biological search, called hBOSOS. The global search capability of BOA and the local search capability of SOS are combined to make the proposed hBOSOS robust and effective.

In addition to improving the BOA, researchers have carried out a wide range of applications of BOA. Arora et al. [42] applied BOA to feature selection. Jalali et al. [43] applied BOA to train artificial neural network. Wang et al. [44] used butterfly optimization algorithm to optimize the extreme learning machine technology and applied it to port throughput prediction. Kisi et al. [45] used BOA to forecast drought in a semi-arid environment. Priyadarshini et al. [46] used BOA to optimize capacitated vehicle routing problem (CVRP). Abdul-Rashid et al. [47] used BOA to optimize the parameters of a designed Lead-Lag Controller. El Hajjami et al. [48] used BOA optimal PID control of an autonomous vehicle. Sharma et al. [49] used BOA/ABC for node localization in acoustic sensor networks. Du et al. [44] used butterfly optimization algorithm to optimize the ELM technology for container throughput prediction. These successful applications are due to BOA's advantages over other optimization methods. That is to say, BOA has a few parameters to be instantiated, and no preprocessing stage is required before the main body of the BOA [40]. Moreover, the algorithm is simple and has strong expansibility, which attracts researchers to expand and improve it and apply it in more and more complex fields.

Another efficient metaheuristic algorithm is flower pollination algorithm (FPA), developed by Yang. FPA belongs to bio-inspired algorithms that simulate flower pollination behavior in nature [7]. In recent years, there have been a large number of improved versions of the pollination algorithm, and it has been applied to solve practical optimization

problems. The improved FPA versions include: Wang et al. [50] proposed the flower pollination algorithm with dimension by dimension improvement; Zhao et al. [51] proposed a complex encoding flower pollination algorithm; Zhou et al. [52] proposed the elite opposition-based flower pollination algorithm. Nabil et al. [53] proposed a modified flower pollination algorithm. Singh et al. [54] proposed an extended version of flower pollination algorithm. Lei et al. [55] proposed wind-driven flower pollination algorithm. Pauline et al. [56] proposed an improved flower pollination algorithm with chaos theory. At the same time, FPA has been successfully applied to solve various practical optimization problems, mainly in the following areas: cloud computing [57], data clustering [58–60], wireless sensor networks (WSNs) optimization [61–63], graph coloring problem [64], neural networks training [65, 66], economic load dispatch problem [67–70], ratios optimization problems (ROPs) [71], vehicle path planning problem [72, 73] and medical image segmentation [74], etc.

BOA has been applied to many aspects, but in terms of the algorithm itself, BOA is easy to fall into the local optimal solution in the early search process; one of the reasons is that the optimal solution is not fully utilized. Moreover, it was observed that random selection of exploration and exploitation phases based on the selected value of switching probability sometimes causes the BOA to lose direction and move away from the global best solution [37]. Therefore, in order to effectively improve the algorithm in the exploration and exploitation capacity and better balance the algorithm in the two stages of switching, in this paper, a hybrid butterfly optimization algorithm and flower pollination base on mutualism mechanism were proposed, which is called MBFPA.

Mutualism is the most common and important inter-specific relationship in ecosystem, and ecosystem cannot exist without mutualism. In nature, flowers send out fragrance to attract butterflies to spread pollen, which makes flowers bear fruit. Butterflies can get nectar and habitat from flowers. The processes of butterfly foraging and flower pollination are a mutually beneficial process. Inspired by the mutualism of different species in nature, a hybrid metaheuristic algorithm using butterfly and flower pollination base on mutualism mechanism was proposed. At the same time, the adaptive switching probability is introduced so that the algorithm can effectively balance the exploration and exploitation stages in the process of operation.

The primary contributions of this paper are summarized as follows:

1. A hybrid metaheuristic algorithm using Butterfly and Flower Pollination base on mutualism mechanism is proposed.
2. The exploration capability of the BOA and FPA is retained, and the exploitation capability is increased

through the mutualism mechanism; the dynamic switching probability balance exploitation and exploration ratio is introduced.

3. To fully test the effectiveness of MBFPA, several performance aspects including accuracy, convergence and statistics are evaluated by using 49 complex benchmark functions.
4. The proposed MBFPA algorithm has been utilized to solve six constrained engineering optimization problems, such as (a) three-bar truss design problem; (b) multi-plate disc clutch brake design; (c) pressure vessel design problem; (d) welded beam design problem and (e) speed reducer design.

The rest of this paper is organized as follows: Sect. 2 briefly introduces the BOA and FPA. Section 3 introduces a hybrid butterfly optimization algorithm and flower pollination algorithm base on mutualism mechanism (MBFPA). Section 4 describes the theoretical comparison with other algorithms. Section 5 describes simulation experiments and result analysis. Finally, our conclusions and ideas for future work are presented in Sect. 6.

2 Butterfly optimization algorithm and flower pollination algorithm

2.1 Butterfly optimization algorithm

The butterfly optimization algorithm [13] is a new metaheuristic algorithm. The algorithm is mainly based on the foraging strategy of butterflies, which utilize their sense of smell to determine the location of nectar or mating partner. In order to find the source of nectar, butterflies use sense receptors which are used to smell and these receptors are scattered over butterfly's body parts like legs, palps, antennae, etc. [37]. In the BOA, it is assumed that each butterfly can release its own fragrance, which is related to its fitness value. The foraging of butterflies can be divided into two situations: when the butterfly can feel the fragrance of the best butterfly in the search space, it will move towards the best butterfly, which is called the global search stage of BOA. When the butterfly can't detect the smell of other butterflies, it will move forward randomly. This stage is called local search. The process of BOA algorithm is described by the following three rules [13]:

1. The fragrance is formulated as a function of the physical intensity of stimulus as follows:

$$f_i = cI^a \quad (1)$$

where f_i is the perceived magnitude of fragrance, how stronger the fragrance is perceived by other butterflies, c is the sensory modality, I is the stimulus intensity and a is the power exponent depended on modality, which accounts the varying degree of absorption.

- In global search phase, the butterfly takes a step towards the fittest butterfly/solution g^* which can be represented as:

$$X_i^{t+1} = X_i^t + (r^2 \times g^* - X_i^t) \times f_i \quad (2)$$

X_i^t is the solution vector X_i of the i th butterfly in the t iteration. g^* represents the best solution currently found. Fragrance of i th butterfly is represented by f_i and r is a random number in $[0,1]$.

- Local search phase can be represented as:

$$X_i^{t+1} = X_i^t + (r^2 \times X_j^t - X_k^t) \times f_i \quad (3)$$

where X_j^t and X_k^t are the j th and k th butterflies randomly selected in the solution space. The switch probability p is used in BOA to switch between global search and local search. BOA is presented in Algorithm 1.

Algorithm 1 Pseudo code for Butterfly optimization algorithm

1: Objective $f(X_i)$, $X_i = (x_1, x_2, \dots, x_d)$, d =number of dimensions

2: Generate initial population of n Butterflies x_i ($i = 1, 2, \dots, n$)

3: Stimulus Intensity I_i at X_i is determined by $f(X_i)$

4: Define sensor modality c power exponent a and switch probability p

5: **while** stopping criteria not met do

6: **for each** butterfly bf in population do

7: Calculate fragrance for bf using Eq. (1)

8: **end for**

9: Find the best bf

10: **for each** butterfly bf in population do

11: Generate a random number r from $[0,1]$

12: **if** $r < p$ then

13: Move towards best butterfly/solution using Eq. (2)

14: **else**

15: Move randomly using Eq. (3)

16: **end if**

17: **end for**

18: Update the value of a

19: **end while**

2.2 Flower pollination algorithm

Flower pollination algorithm (FPA) [7] is a new metaheuristic swarm intelligence optimization algorithm proposed by Xinshe Yang in 2012. This optimization algorithm simulates the pollination process of plant flowers in nature, mainly including cross-pollination and self-pollination of flowers. In the process of cross-pollination, the flight behavior of the propagator (butterfly, bee, etc.) obeys Levy flight distribution. In the process of self-pollination, the mature pollen of plants spreads to their own flowers or different flowers of the same type of plants. The process of FPA algorithm is described by the following four rules [7]:

- Flower constancy can be considered as the reproduction probability is proportional to the similarity of two flowers involved.
- In the implementation of the algorithm, the conversion between local and global pollination mechanisms is controlled by the value of conversion probability $p \in [0,1]$.
- In the process of biological cross-pollination, the Propagator's flight obeys Levy flight for global pollination. The formula is described as follows:

$$X_i^{t+1} = X_i^t + Levy(\lambda)(X_i^t - g^*) \quad (4)$$

where X_i^{t+1} and X_i^t represent the position of pollen individual i in the $t + 1$ and t generations, g^* represents the position of the optimal flower and/or a pollen gamete the current population and represents Levy the step factor obeying the Levy distribution, as shown in the following formula:

$$L \sim \frac{\lambda \Gamma(\lambda) \sin(\pi \lambda / 2)}{\pi} \frac{1}{S^{1+\lambda}}, (S \geq S_0 > 0) \quad (5)$$

where $\Gamma(\lambda)$ is the standard gamma function, $\lambda = 1.5$.

- Abiotic self-pollination can be regarded as the local pollination stage in the algorithm, which is expressed as

$$X_i^{t+1} = X_i^t + \varepsilon(X_j^t - X_k^t) \quad (6)$$

where X_j^t and X_k^t represent the position of two different pollens in the same kind of plants. They are randomly selected two individuals; ε is the variable of uniform distribution in $[0, 1]$. FPA is presented in Algorithm 2.

Algorithm 2 Pseudo code for Flower pollination algorithm

```

1: Objective  $f(X_i), X_i = (x_1, x_2, \dots, x_d)$ ,  $d$  =number of dimensions
2: Generate initial population of n flowers/pollen  $x_i (i=1,2,\dots,n)$ 
3:
4: Define switch probability  $p$ 
5: while stopping criteria not met do
6: for  $i=1:n$ (all n flowers in the population)
7:   Generate a random number  $r$  from  $[0,1]$ 
8:   if  $r < p$  then
9:     generate a (d-dimensional) step vector  $L$ , which obeys the levy distribution
10:    Global pollination using Eq. (4)
11:   else
12:     generate a random number  $\varepsilon$  from  $[0,1]$ 
13:     Local pollination using Eq.(6)
14:   end if
15:   Evaluate the new solutions
16:   if new solutions are better, update them in the population
17: end for
18: Find the best solution
19: end while

```

3 Hybrid butterfly optimization algorithm and flower pollination algorithm base on mutualism mechanism

As already discussed, since the butterfly optimization algorithm was proposed, researchers have made different attempts to better improve the algorithm's exploration and exploitation capabilities: Arora and Singh proposed BOA/DE [35], BOA/ABC [34], balanced the algorithm's ability to exploitation and exploration. Because the two different algorithms are mixed, the advantages of the original algorithm can be retained. In addition, the addition of two different renewal mechanisms will inevitably increase the diversity of the population. Sharma and Sushmita [40] effectively improved the algorithm exploitation ability by adding the SOS algorithm to the exploitation stage of the original butterfly optimization algorithm. A good algorithm should be able to take into account two aspects. When the exploitation phase of an algorithm is dominant, the early population of the algorithm will concentrate on the best individual attachments early, leading to the loss of population diversity.

During the algorithm update process, If too much consideration is given to the exploration capability of the algorithm, the exploration with low accuracy may lose the optimal solution and slow convergence rate.

Therefore, how to balance the two aspects of the algorithm becomes the key.

Thus, the success of a metaheuristic method on a given optimization problem is defined by its ability to provide a good balance between the exploration and exploitation. The exploration defines the global search ability of the algorithm, whereas the exploitation is the ability to find the optimum around a near-optimal solution, which can also be considered as the local search ability [75].

Therefore, in the current study, in order to deal with this imbalance between exploration capability and exploitation capability, BOA and FPA were adopted to mix through mutualism. The combination of diverse metaheuristics can lead to new exciting approaches since the hybridization can be used to get the advantage of different metaheuristics [76]. Here, the first step is to divide the entire population into two subpopulations: butterflies and flowers. Independent evolution between subpopulations can gain the advantages of both algorithms while increasing the diversity of the entire population. At the same time, the dynamic switching probability is introduced to balance the reasonable distribution of exploration and exploitation. The second stage is called mutualism stage. In this stage, the individual butterfly should randomly select a flower for mutualism, and the flower should do the same, so as to increase the exploitation capability of the algorithm.

Mutualism refers to two kinds of organisms living together, which are mutually beneficial. After the two are separated, life will be greatly affected, even death. Among them, insects, birds, mammals and many other creatures serve for pollination and seed transmission of flowering plants. Plants provide them with nectar and fruits in return. Without symbiosis, most plants would not survive. An example of reciprocity is the relationship between the butterfly and the flower, and the butterfly looks for food/nectar in the flower cluster for the butterfly to maintain its own survival, which is also beneficial to the flower in this activity, because the butterfly's foraging distributes pollen in the process, which is beneficial to the pollination of plants. It's good for both sides of life.

In order to simulate the mutualism between butterfly and flower, the mutualism stage of symbiotic organisms search (SOS) is introduced. In 2014, Cheng and Prayogo [12] proposed SOS, which simulates the interaction between

organisms in the ecosystem. In the search space, each individual of different species is regarded as a candidate solution. The algorithm randomly initializes n organisms and generates an ecosystem. On this basis, individuals are updated through mutualism, commensalism and parasitism. SOS has good exploitation capabilities with the processes of mutualism and commensalism. SOS uses best solution as a reference point that might help in exploiting the neighborhood solutions of the current best solution [77].

This paper introduces the mutualism phase in the SOS algorithm. The renewal formula of SOS mutualism stage is introduced as follows [12]:

$$Mutual_{agent} = (X_i^t + X_j^t) / 2 \tag{7}$$

$$X_i^{t+1} = X_i^t + rand[0, 1] \times (g^* - Mutual_{agent} \times BF1) \tag{8}$$

$$X_j^{t+1} = X_j^t + rand[0, 1] \times (g^* - Mutual_{agent} \times BF2) \tag{9}$$

where $Mutual_{agent}$ expressed the relationship between the two organisms in the t generation x_i^t and x_j^t . $rand[0,1]$ is a vector of random numbers. g^* is the best organism in the population, and BF1 and BF2 are the interest factors of randomly generated 1 or 2. These factors represent the favorable degree of interaction between two organisms.

In addition, the BOA algorithm uses fixed switching probability in the whole search process. However, the reasonable process should be: search in the global scope in the early stage, and increase the mining intensity in the local scope in the later stage to increase the accuracy of the solution. Therefore, in order to effectively balance the two stages of exploration and exploitation, the dynamic switching probability is used, as shown in the following formula:

$$p = 0.8 - 0.1 \times (Max_iter - itrer) / Max_iter \tag{10}$$

where Max_iter is the maximum number of iterations and $itrer$ is the current number of iterations.

The flowchart of the proposed approach is described in Fig. 1, and pseudocode for the mutualism butterfly flower pollination algorithm is presented in “Algorithm 3”:

Algorithm 3 Pseudo code for the proposed MBFPA algorithm

```

Input: Objective function  $f(x), x = (x_1, x_2, \dots, x_d)$ ,
 $d = \text{no. of dimensions}$ 
 $Max\_iter = \text{maximum number of iteration}$ 
Initialize a population of n agents with random  $x_i (i = 1, 2, \dots, n)$ 
Find the best solution  $g^*$  in the initial population
Divide n individuals into two subpopulations,  $S_1$  and  $S_2$ .
 $S_1$  represents butterfly subpopulation,  $S_2$  represents the flower subpopulation
Define sensor modality  $c$ , power exponent  $a$  and switch probability  $p$ 
While stopping criteria do not met do
  For each agent in the population do
    if  $x_i^t \in S_1$ 
      if  $rand < p$ 
        Calculate fragrance for agent using Eq.(1)
        Perform global search using (2) and obtain new solution  $x_{new}^t$ 
      else
        Perform local search using (3) and obtain new solution  $x_{new}^t$ 
      end
    else
      if  $rand < p$ 
        Perform global pollination using (4) and obtain new solution  $x_{new}^t$ 
      else
        Perform local search using (5) and obtain new solution  $x_{new}^t$ 
      end
    End
    Calculate fitness value of the new solution
    if  $f(x_{new}^t) < f(x_i^t)$ 
       $x_i^t = x_{new}^t$ 
    end
    if  $x_i^t \in S_1$ 
      Randomly select an individual  $X_j^t, X_j^t \in S_2$ 
    else
      Randomly select an individual  $X_j^t, X_j^t \in S_1$ 
    end
    Determine mutual relationship Mutual Vector by Eq.(7)
    Update the current individual based on their mutual relationship
      according to Eq. (8) and (9);
    Evaluate the new solutions
    If new solutions are better, update them in the population
  end
  Find the current best solution  $g^*$ 
end

```

4 Theoretical comparison with other algorithm

In the BOA (Arora and Singh), the butterfly optimization algorithm is conceptually compared with other algorithms, namely ABC, CS, DE, FA, GA, MBO and PSO. Based on the comparison of the BOA, m-MBOA makes a comprehensive conceptual comparison among SOS, Jaya, improved BOA, chaotic BOA (CBOA), modified BOA, mutated BOA(BOA-C), BOA/DE and BOA/ABC.

The difference between the proposed algorithm and m-MBOA is that m-MBOA only introduces the symbiosis stage unilaterally, which increases the exploitation ability of the algorithm, and the symbiosis is based on the symbiosis between species. The introduction of the proposed algorithm in the symbiosis phase is not only between species, but also through the interaction between butterflies and flowers. Additionally, the introduction of dynamic switching probability can better balance the ratio of exploitation and exploration within the subpopulation.

TLBO is an algorithm based on human behavior, and its inspiration comes from the teaching and learning process in the classroom. The iterative evolution process of TLBO includes the teaching phase and the learning phase. To enhance the average knowledge level of the class, learners improve their knowledge levels by learning from the teacher in the teaching phase, and they also improve their knowledge levels by learning interactively from another learner selected randomly in learning phase [30]. MBFPA is a hybrid algorithm of butterfly optimization algorithm and flower pollination algorithm, inspired by the mutually beneficial relationship between butterflies and flowers in nature. Through the combination of the two algorithms, the exploration ability of butterfly algorithm and flower pollination algorithm is preserved. Across the introduction of SOS symbiosis stage, the exploitation ability of the algorithm can be enhanced and the probability of losing the optimal solution of the algorithm can be reduced.

GSA can be considered as physics-based metaheuristic search algorithm. At the beginning of the algorithm, each individual is given a mass, and the law of gravity between two objects is used to guide the motion optimization of each particle to search for the optimal solution. Superposition of the gravitational forces, dependency to the distance and the relation between mass values and fitnesses make this algorithm unique [78]. The proposed algorithm is a metaheuristic algorithm that simulates the mixed symbiotic relationship between two organisms in nature, mainly including two basic algorithms of BOA and FPA, which, respectively, simulate the foraging of butterfly and the pollination of flower, and the two subgroups switch through p for exploration and exploitation. At the same time, the

increased mutualism stage enables the exchange of information between subgroups, reducing the possibility of entering the local optimal solution.

Hbosos is a hybrid algorithm of BOA and SOS. By combining BOA and SOS, the algorithm can retain the advantages of the two algorithms to the greatest extent. The proposed algorithm does not integrate the SOS algorithm, but improves the algorithm development ability by introducing the mutually beneficial symbiosis stage of SOS.

Li, Guocheng et al. proposed an improved butterfly optimization algorithm (BOA) using the cross-entropy method. The cross-entropy (CE) method was developed by Rubinstein [79] in 1997 to estimate the probability of rare events in complex random networks. This paper embeds the CE method into the BOA to obtain a good balance between exploration and exploitation and improve the BOA's global search capability. The proposed algorithm is a hybrid algorithm based on two kinds of biological mutualism mechanisms in nature. It enhances the ability of algorithm development by adding mutualism mechanism. In the value of switching probability, adaptive dynamic switching probability is used.

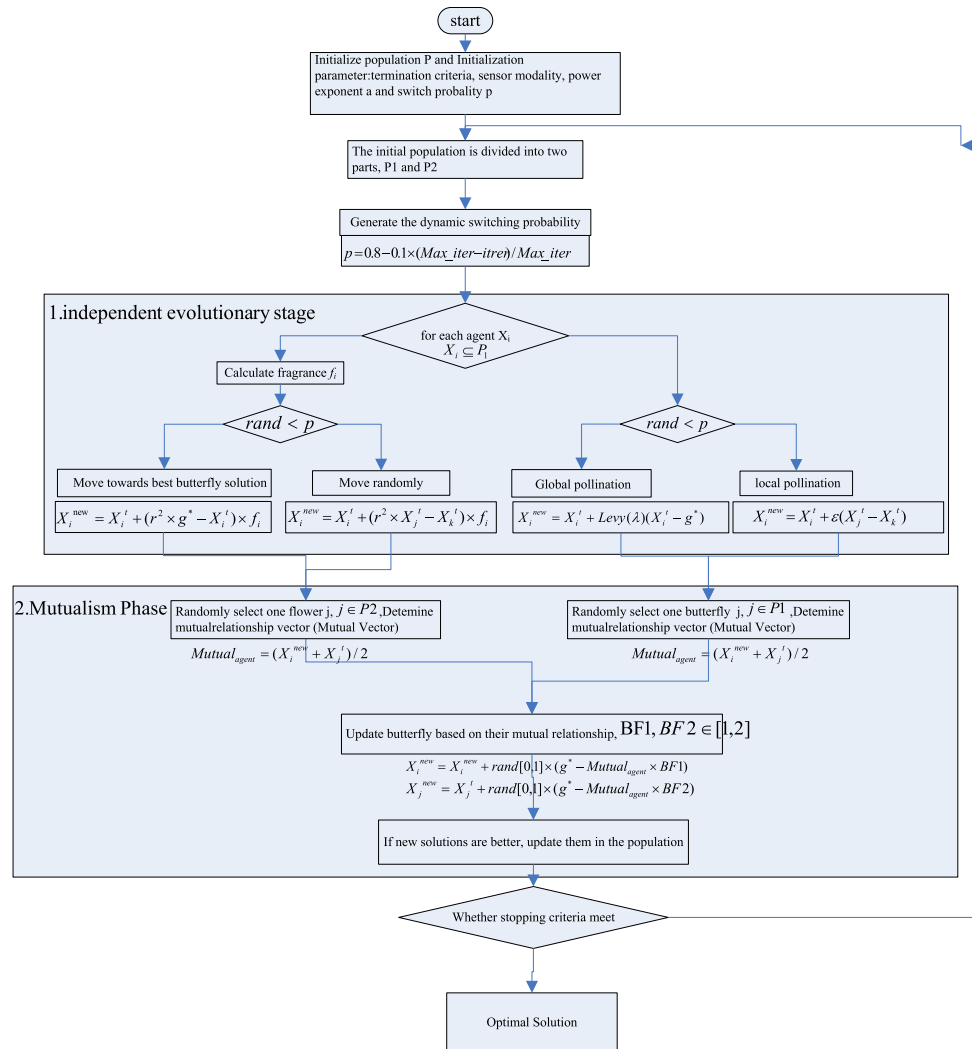
Arora and Singh proposed Learning automata-based butterfly optimization algorithm. Learning automata have been embedded in BOA in which a learning automaton takes the role of configuring the behavior of a butterfly in order to create a proper balance between the process of global and local search [41], by introducing the adaptive dynamic switching probability, which is a simpler and more effective method.

5 Simulation experiments and result analysis

In order to verify the effectiveness of the proposed algorithm, as shown in Table 1, 49 different benchmark functions are tested, including function name, type, dimension, search scope, formulation and optimal value. Generally speaking, reference functions can be divided into two categories: unimodal function and multimodal function. Different types of test functions have different characteristics. Among them, the single-peak test function is used to verify the excavation ability. Multimodal test functions have many local optimal solutions, which is helpful to test the exploration ability of the algorithm and the ability to avoid the optimal value.

The algorithm is implemented in MATLAB R2018b. Experiments are performed on a PC with a 3.30 GHz, Intel(R) Core(TM) i5 CPU, System type 64 bit, Windows 10 operating system. To increase reliability and generate statistically significant results, each function was run 30 times in this validation test. The mean and standard deviation of the proposed algorithm and other algorithms for comparison were recorded.

Fig. 1 Flowchart of the proposed algorithm



5.1 Comparison with basic BOA and enhanced BOA

In order to evaluate the performance of the proposed algorithm compared with BOA [13] and enhanced BOA (m-MBOA) [40], it is worth noting that in the comparison experiment, the parameter Settings of the proposed algorithm are the same as those of the other two algorithms in the paper. In this comparison, the number of population is fixed at 50 and the maximum number of iterations is fixed at 10,000. The simulation results include average results and corresponding standard deviation. In the simulation of all reference functions, sensory modality (c) is 0.01 and power exponent (a) is increased from 0.1 to 0.3.

From the test results shown in Table 2, it can be seen that under the same test conditions, $f_5, f_{14}, f_{20}, f_{25}$ and f_{28} are superior to the other two algorithms, and other test functions

can achieve the same accuracy as the other two algorithms. It can be concluded that the proposed algorithm can get high-precision results; in addition, it has better results in multiple functions.

5.2 Comparison with improve FPA

In order to compare the performance of the algorithm with other improved flower pollination algorithms, it is compared with dimensional evolution FPA (MFPA) [50] and bee FPA (BPFPA) [80]. In the comparison experiment, the population size n is set to 30 and the number of iterations is 500 generations. The switching probability of MFPA and BPFPA is set to 0.8, and the parameter setting of the proposed algorithm is consistent with the previous section.

From the test results shown in Table 3, the results compared in some test functions all reached the same accuracy. Except in f_{10} , f_{25} was less effective than MFPA, and the rest of the test results were better than the comparison algorithm. Therefore, on the whole, the proposed algorithm has better results compared with the two improved FPA (Table 4).

5.3 Comparison with other basic metaheuristic methods

To verify the results of the proposed algorithm, the other six state-of-the-art metaheuristic algorithms are employed: differential evolution (DE), flower pollination algorithm (FPA), particle swarm optimization (PSO), gravitational search algorithm (GSA), symbiotic organisms search (SOS) and teaching learning-based optimization (TLBO). These algorithms have been widely used in various fields. In the current experiment, the population size and maximum iteration of all algorithms are set to 30 and 500, respectively. In 30 independent operations, Table 5 shows function results performance statistics, Table 6 shows the mean and standard deviation of the proposed algorithm and other algorithms in solving the benchmark function, and the bold numbers represent the relatively best values of the compared algorithms. The specific parameter values of the comparison algorithm are shown in Table 4.

From the statistical data, Table 6 shows that the proposed algorithm fails to get satisfactory solutions in functions f_{10} , f_{14} , f_{20} , f_{26} , f_{28} , f_{43} , f_{46} , f_{48} and f_{49} . The functions of f_{10} , f_{14} , f_{20} , f_{28} , f_{48} and f_{49} are far from the theoretical values. Compared with the results obtained by other algorithms in these functions, the DE gets relatively better results in f_{10} and f_{47} . SOS gets relatively better solutions in f_{10} , f_{20} , f_{47} , f_{48} and f_{49} ; the functions of f_{49} are far from the theoretical values. PSO get relatively better solutions in function f_{28} . TLBO gets relatively better solutions in f_{39} , f_{42} , f_{47} , but these values are still different from the theoretical values. From the overall perspective, among the 49 benchmark functions, the proposed algorithm can find the theoretical optimal value among the 27 test functions. However, the number of theoretical values of DE, FPA, PSO, GSA, SOS and TLBO is 16, 8, 6, 4, 15 and 15, respectively. Meanwhile, Table 5 shows the number of occasions where the mean performance of MBFPA is better than, equality and worse than other comparison algorithms. From this table, it can be observed that the proposed algorithm performs better than DE, FPA, PSO, GSA, SOS and TLBO in 25, 39, 40, 44, 22 and 25 benchmark functions, respectively, equality results are seen in 15, 8, 5, 3, 18 and 18, and worse results are obtained in 3, 2, 3, 1, 4 and 2. Through the comparison of the experimental results and data, we can see that in most of the test functions, the

proposed algorithm can get a more satisfactory solution, and has a strong competitiveness with the compared algorithm.

In addition, the performance of the algorithm is statistically tested. Wilcoxon nonparametric statistical test [80] and Friedman rank test [81] were conducted in this paper. Wilcoxon's non-parametric statistical test returns a parameter call p-value [80]. When p value is less than 0.05, there is a significant difference between the two algorithms in solving the problem; when $p > 0.05$, there is no significant difference between the two algorithms in solving the problem. From the p -value test results in Table 7, except for functions f_2 , f_4 , f_6 , f_9 , f_{13} , f_{15} , f_{21} , f_{22} , f_{31} , f_{36} and f_{44} , most of the algorithms can converge to the theoretical value, resulting in no significant difference in the test results of these functions; for the remaining functions, in function f_{16} and f_{41} the proposed algorithm has no significant difference with SOS; the proposed algorithm has no significant difference with PSO in function f_{22} , f_{37} , f_{42} and f_{49} ; the proposed algorithm has no significant difference with TLBO in function f_{22} , f_{26} , f_{37} , f_{39} , f_{41} , f_{48} . There are significant differences in other functions. Therefore, from the p value test results, this further shows that the proposed algorithm has superior performance.

Table 8 presents the ranks obtained by Friedman rank test from the mean performances of the algorithms for each benchmark functions. As can be seen from Table 8, MBFPA has the smallest grade, which indicates that MBFPA's performance is better than the comparison algorithm (Table 9).

In order to compare the advantages of the algorithm in terms of convergence speed, Figs. 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13 and 14 show the convergence curve of the proposed algorithm and DE, FPA, PSO, GSA, SOS, TLBO on 14 benchmark functions. Among all the fitness value convergence graphs, their convergence curves are based on the results of 30 times of independent operation of 7 algorithms, and all the convergence curves are drawn with the average value. As shown in Figs. 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14 and 15, f_{11} , f_{14} , f_{24} , f_{30} and f_{33} can reach the fastest convergence speed and can find the global optimal solution. In f_6 functions, the fastest convergence speed is not achieved, but still can have a better ranking and ultimately can find the global optimal solution; f_1 , f_3 , f_5 , f_7 , f_{12} , f_{19} , f_{23} function in the early convergence speed is not the first, but in the later search, it reaches the highest convergence rate and continues to converge to the highest accuracy. Although the convergence effect of some functions cannot reach the fastest convergence rate in the early stage, and the convergence rate of f_6 is lower than that of SOS, overall speaking, the convergence rate of the proposed algorithm is better than that of most other comparison algorithms.

5.4 Compared with the hybrid metaheuristic algorithm

In addition to the comparison with the basic metaheuristic algorithm, the other five hybrid metaheuristic algorithms are employed: hybrid algorithm of particle swarm optimization and Grey wolf optimizer (PSOGWO) [82], sine cosine crow search algorithm (SCSCA) [83], hybrid firefly and particle swarm optimization algorithm (HFPSO) [82], modified flower pollination algorithm (MFPA) [53] and hybrid whale optimization algorithm based on modified differential evolution (MDE-WOA) [84].

In the current experiment, the population size and maximum iteration number of all algorithms are set to 30 and 500, respectively. Among the 30 independent operations, Table 10 shows the performance statistics of the results of the function, and Table 11 shows the mean and standard deviation of the proposed algorithm and other algorithms in solving the benchmark function. Bold numbers represent the relative optimal values of the compared algorithms. The specific parameter values of the comparison algorithm are shown in Table 9.

According to the performance statistics in Table 10, among the 49 test results of MBFPA, 48 test results were better than PSOGWO. In the SCSCA results, the test results of f_{10} , f_{14} , f_{20} , f_{26} , f_{28} and f_{34} are better than the proposed algorithm, but there is a large gap with the theoretical value. In other test functions, it can be seen from Table 10 that the test results of the proposed algorithm are better than those of SCSCA in 38 test functions and achieve the same effect in 5 test results. In the HFPSO results, the HFPSO test results are better than the proposed algorithm in the test functions f_{10} , f_{20} , f_{43} and f_{49} , but f_{20} and f_{49} are far from the theoretical values. In the remaining test functions, the proposed algorithm is better than the HFPSO in 37 test results and gets the same value in eight test function results. In the test results of MFPA, the proposed algorithm is not better than MFPA in 11 test functions, but better than the comparison algorithm in 24 test function results, and achieves the same effect in 14 test functions. In the test results of MDE-WOA, the results of functions f_{14} , f_{20} , f_{28} , f_{34} , f_{43} and f_{46} are better than the proposed algorithm, but it can be seen from Table 10 that the proposed algorithm is better than MDE-WOA in the 38 test results. Therefore, through the comparison of experimental results and data, we can see that in most of the test functions, the algorithm proposed in this paper can get more satisfactory solutions and has a strong competitiveness compared with the comparison algorithm.

In addition, from the p -value test results in Table 12, there is no significant difference in the test results of these functions, except that the comparison algorithm reaches

the theoretical value in individual test results; for most of the other test functions, p -value is less than 0.05. Table 13 presents the ranks obtained by Friedman rank test from the mean performances of the algorithms for each benchmark functions. It can be seen from Table 13 that MBFPA has the smallest grade, which indicates that MBFPA's performance is better than the hybrid metaheuristic algorithm compared.

Figures 16, 17, 18, 19, 20, 21, 22 and 23 illustrate the convergence of the fitness values of PSOGWO, SCSCA, MFPA, HFPSO, BPFPA and MDE-WOA. These convergence graphs are based on the results of 30 independent runs of the six algorithms. From these figures, it can be clearly seen that MBFPA obtains the global optimal value faster than the other four algorithms. These experimental results demonstrate that MBFPA, which reflects its strong global search capability.

5.5 Results analysis

Now, why is MBFPA so effective? It mainly includes the following three aspects:

1. The introduction of dynamic switching probability perfectly balances the distribution of exploitation and exploration.
2. Because of the mixture of BOA and FPA, the exploration capability of the two algorithms is well reserved.
3. Through the introduction of mutualism phase, the mutualism of individuals enhances the exploitation ability of the algorithm.

In particular, the butterfly optimization algorithm controls the switch between exploitation and exploration by transforming probability p . Because every step of exploitation and exploration switch is judged by $\text{rand} < p$, it may cause the individual who has been in the vicinity of the optimal solution to turn to exploration and lose the optimal solution. In addition, the BOA does not directly use the best solution as the reference point, but through $r^2 \times g^*$ as the reference point, where r is a random number and g^* is the best solution in the population, which reduces the guiding role of the best solution. Although it can reduce the probability of falling into the local optimum prematurely, it has a great limit on the exploitation ability of the algorithm. In the improved algorithm, the dynamic switching probability is introduced to distribute the proportion of development and exploration more reasonably. In the increasing symbiosis mechanism, on the one hand, the optimal position is used as the reference update position, which increases the ability of individuals to find the optimal solution around the near optimal solution;

Table 1 Benchmark functions (dim: dimensions, M: multimodal, N: non-separable, U: unimodal, S: separable)

No	Function	Type	dim	Search space	Formulation	f_{min}
f_1	Sphere	M,S	30	$[-100,100]$	$f(x) = \sum_{i=1}^n x_i^2$	0
f_2	Beale	U,N	2	$[-4.5,4.5]$	$f(x) = (1.5 - x_1 + x_1x_2)2 + (2.25 - x_1 + x_1x_2^2) + (2.625 - x_1 + x_1x_2^3)^2$	0
f_3	Cigar	U,N	30	$[-100,100]$	$f(x) = x_1^2 + 10^6 \sum_{i=2}^n x_i^2$	0
f_4	Step	U,S	30	$[-100,100]$	$f(x) = (\lfloor x_i + 0.5 \rfloor)^2$	0
f_5	Quadraic	U,S	30	$[-1.28,1.28]$	$f(x) = \sum_{i=1}^n ix_i^4 + rand(0, 1)$	0
f_6	Bohachevsky	M,N	2	$[-100,100]$	$f(x) = x_1^2 + 2x_2^2 - 0.3 \cos(3\pi x_1) - 0.4 \cos(4\pi x_2) + 0.7$	0
f_7	Ackley	M,N	30	$[-100,100]$	$f(x) = -20 \exp \left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2} - \exp \left(\frac{1}{n} \sum_{i=1}^n \cos 2\pi x_i \right) \right) + 20 + e$	0
f_8	Griewank	M,N	30	$[-100,100]$	$f(x) = \frac{1}{4000} \sum_{i=1}^n (x_i^2) - \prod_{i=1}^n \cos \left(\frac{x_i}{\sqrt{i}} \right) + 1$	0
f_9	Levy	M,S	2	$[-10,10]$	$f(x) = \sin^2(3\pi x_1) + (x_1 - 1)2[1 + \sin^2(3x_2)] + (x_1 - 1)^2 \times [1 + \sin^2(2\pi x_2)]$	0
f_{10}	Michalewiz	M,S	10		$f(x) = - \sum_{i=1}^n \sin(x_i) \left[\sin \left(\frac{ix_i^2}{\pi} \right) \right]^{2m}, m = 10$	- 0.966
f_{11}	Rastrigin	M,S	30	$[-5.12,5.12]$	$f(x) = \sum_{i=1}^n x_i^2 - 10 \cos(2\pi x_i + 10)$	0
f_{12}	Alpine	M,S	30	$[-10,10]$	$f(x) = \sum_{i=1}^n x_i \sin(x) + 0.1x_i $	0
f_{13}	Schaffer	M,N	2	$[-100,100]$	$f(x) = (x_1^2 + x_2^2)^{0.25} [50(x_1^2 + x_2^2)0.1 + 1]$	0
f_{14}	Rosenbrock	U,N	30	$[-30,30]$	$f(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	0
f_{15}	Easom	M,S	2	$[-100,100]$	$f(x) = - \cos(x_1) \cos(x_2) \exp(-(x_1 - \pi)^2 - (x_2 - \pi)^2)$	- 1
f_{16}	Shubert	M,S	2	$[-100,100]$	$f(x) = \left(\sum_{i=1}^5 i \cos((i+1)x_1 + i) \right) \left(\sum_{i=1}^5 i \cos((i+1)x_2 + i) \right)$	- 186.7309
f_{17}	Schwefel 1.2	U,N	30	$[-100,100]$	$f(x) = \sum_{i=1}^n \left(\sum_{j=1}^i x_j \right)^2$	0
f_{18}	Schwefel 2.21	U,S	30	$[-100,100]$	$f(x) = \max_i \{ x_i , 1 \leq i \leq n\}$	0
f_{19}	Schwefel 2.22	U,N	30	$[-10,10]$	$f(x) = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $	0
f_{20}	Schwefel 2.26	M,S	30	$[-500,500]$	$f(x) = \sum_{i=1}^n x_i \sin(x) + 0.1x_i $	- 418.982*n
f_{21}	Booth	U,N	2	$[10, 10]$	$f(x) = (x_1 + 2x_2 - 7)^2 + (2x_1 + x_2 - 5)^2$	0
f_{22}	Goldstein price	M,N	2	$[-2,2]$	$f(x) = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] \times [30 + (2x_1 - 3x_2)^2(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$	3
f_{23}	Matyas	U,N	2	$[-10,10]$	$f(x) = 0.26(x_1^2 + x_2^2) - 0.48x_1x_2^2$	0
f_{24}	Powell	U,N	30	$[-100,100]$	$f(x) = \sum_{i=1}^{n/4} (x_{4i-3} + 10x_{4i-2})^2 + 5(x_{4i} + 10x_{4i})^2 + (x_{4i-2} + 2x_{4i-1})^4 + 10(x_{4i-3} + 10x_{4i})^4$	0
f_{25}	Power sum	M,N	4	$[-10,10]$	$f(x) = \sum_{i=1}^n \left[\left(\sum_{k=1}^n x_k^i \right) - b_i \right]$	0

Table 1 (continued)

No	Function	Type	dim	Search space	Formulation	f_{min}
f_{26}	Shekel 4.5	U,S	4	[0,10]	$f(x) = -\sum_{i=1}^5 [(x - a_i)(x - a_i)^T + c_i]^{-1}$	- 10,1532
f_{27}	Sum square	U,S	30	[- 10,10]	$f(x) = \sum_{i=1}^n ix_i^2$	0
f_{28}	Trid	M,N	30	[- 10,10]	$f(x) = \sum_{i=1}^n (x_i - 1)^2 - \sum_{i=2}^n (x_i x_{i-1})$	-d*(d+4)*(d-1)/6
f_{29}	Zettl	U,N	2	[- 1,5]	$f(x) = (x_1^2 + x_2^2 - 2x_1)2 + 0.25x_1$	- 0.00379
f_{30}	Leon	U,N	2	[- 1.2,1.2]	$f(x) = 100(x_2 - x_1^3)^2 + (1 - x_1)^2$	0
f_{31}	Stepint	U,S	5	[- 5.12,5.12]	$f(x) = 25 + \sum_{i=1}^5 [x_i]$	0
f_{32}	Colville	U,N	4	[- 10,10]	$f(x) = 100(x_1^2 - x_2^2)^2 + (x_1 - 1)^2 + (x_3 - 1)^2 + 90(x_3^2 - x_4)^2 + 10.1(x_2 - 1)^2 + (x_4 - 1)^2 + 19.8(x_2^2 - 1)(x_4 - 1)$	0
f_{33}	Zakharov	U,N	10	[5, 10]	$f(x) = \sum_{i=1}^n x_i^2 + (\sum_{i=1}^n 0.5ix_i)^2 + (\sum_{i=1}^n 0.5ix_i)^4$	0
f_{34}	Dixon-Price	U,N	30	[- 30,30]	$f(x) = (x_1 - 1)^2 + \sum_{i=2}^n i(2x_i^2 - x_{i-1}^2)$	0
f_{35}	Foxholes	M,S	2	[- 65.536, 65.536]	$f(x) = \left[\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^2 (x_i - a_{ij})^6} \right]^{-1}$	0.998
f_{36}	Branin	M,S	2	[5, 10]	$f(x) = (x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6)^2 + 10(1 - \frac{5.1}{4\pi^2})\cos x_1 + 10$	0.398
f_{37}	Six Hump Camel Back	M,N	2	[- 5,5]	$f(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	- 1.03163
f_{38}	Kowalik	M,N	4	[- 5,5]	$f(x) = \sum_{i=1}^{11} \left(ai - \frac{x_1(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4} \right)^2$	0.00031
f_{39}	Perm	M,N	4	[- 4,4]	$f(x) = \sum_{k=1}^n \left[\sum_{i=1}^n (i_k + \beta)(x_i/i^k) - 1 \right]^2$	0
f_{40}	Hartman	M,N	3	[0,1]	$f(x) = \sum_{k=1}^n c_k \exp \left[-\sum_{j=1}^3 a_{ij}(x_j - p_{ij})^2 \right]$	- 3.86
f_{41}	Hartman	M,N	6	[0,1]	$f(x) = \sum_{k=1}^n c_k \exp \left[-\sum_{j=1}^6 a_{ij}(x_j - p_{ij})^2 \right]$	- 3.32
f_{42}	Penalized	M,N	30	[- 50,50]	$f(x) = \frac{\pi}{D} \left\{ 10 \sin^2(\pi y_1) + \sum_{i=1}^{D-1} (y - 1)^2 [1 + 10 \sin^2(\pi y_i)] \right\} + \sum_{i=1}^D u(x_i, 10, 100, 4)$ $y_i = 1 + \frac{x_i + 1}{4}$ $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m, x^i > a \\ 0, -a \leq x_i \leq a \\ k(-x_i - z)^m, x_i < a \end{cases}$	0
f_{43}	Penalized2	M,N	30	[- 50,50]	$f(x) = 0.1 \left\{ \sin^2 \left(3\pi x_1 + \sum_{i=1}^n (x_i - 1)^2 [1 + \sin^2(3\pi x_i + 1)] \right) + \sum_{i=1}^n u(x_i, 5, 100, 4) \right\}$	0
f_{44}	Langerman2	M,N	2	[0,10]	$f(x) = \sum_{i=1}^m c_i \exp \left[-\frac{1}{\pi} \sum_{j=1}^n (x_j - a_{ij})^2 \cos(\pi \sum_{j=1}^n (x_j - a_{ij})^2) \right]$	- 1.08

Table 1 (continued)

No	Function	Type	dim	Search space	Formulation	f_{min}
f45	Langerman5	M,N	5	[0,10]	$f(x) = \sum_{i=1}^m c_i \exp \left[-\frac{1}{\pi} \sum_{j=1}^n (x_j - a_{ij})^2 \cos(\pi \sum_{j=1}^n (x_j - a_{ij})^2) \right]$	- 1.5
f46	Langerman10	M,N	10	[0,10]	$f(x) = \sum_{i=1}^m c_i \exp \left[-\frac{1}{\pi} \sum_{j=1}^n (x_j - a_{ij})^2 \cos(\pi \sum_{j=1}^n (x_j - a_{ij})^2) \right]$	N/A
f47	Fletcher Powell2	M,N	2	$[\pi, \pi]$	$f(x) = \sum_{i=1}^n (A_i - B_i)^2$ $A_i = \sum_{j=1}^n (a_{ij} \sin a_j + b_{ij} \cos a_j), B_i = \sum_{j=1}^n (a_{ij} \sin x_j + b_{ij} \cos x_j)$	0
f48	Fletcher Powell5	M,N	5	$[\pi, \pi]$	$f(x) = \sum_{i=1}^n (A_i - B_i)^2$ $A_i = \sum_{j=1}^n (a_{ij} \sin a_j + b_{ij} \cos a_j), B_i = \sum_{j=1}^n (a_{ij} \sin x_j + b_{ij} \cos x_j)$	0
f49	Fletcher Powell10	M,N	10	$[\pi, \pi]$	$f(x) = \sum_{i=1}^n (A_i - B_i)^2$ $A_i = \sum_{j=1}^n (a_{ij} \sin a_j + b_{ij} \cos a_j), B_i = \sum_{j=1}^n (a_{ij} \sin x_j + b_{ij} \cos x_j)$	0

Table 2 Mean and SD for comparing MBPFA with BOA and m-MBOA on 30 benchmark functions

	Mean			SD		
	MBPFA	m-MBOA [38]	BOA [13]	MBPFA	m-MBOA [38]	BOA [13]
f1	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
f2	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
f3	0.00E+00	0.0000E+00	0.00E+00	0.00E+00	0.0000E+00	0.00E+00
f4	0.00E+00	0.0000E+00	0.00E+00	0.00E+00	0.0000E+00	0.00E+00
f5	5.7759E-06	3.8917E-05	1.45E+00	4.1738E-05	2.9003E-05	1.73E-01
f6	0.00E+00	0.0000E+00	0.00E+00	0.00E+00	0.0000E+00	0.00E+00
f7	8.8818E-16	1.7183E+00	8.8818E-16	0.0000E+00	0.0000E+00	0.0000E+00
f8	0.0000E+00	1.8472E-19	0.00E+00	0.0000E+00	2.6886E-02	0.0000E+00
f9	1.35E-31	4.4108E-01	1.35E-31	6.6809E-47	5.7467E-02	8.9000E-47
f10	- 6.08E+00	N/A	- 5.34E-02	5.1538E-01	N/A	- 5.6092E+00
f11	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
f12	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
f13	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
f14	2.0681E+01	2.8837E+01	5.80E+00	1.5593E+00	0.0000E+00	6.12E-01
f15	- 1.000E+00	- 1.0000E+00	- 1.02E+00	0.0000E+00	0.0000E+00	1.51E-01
f16	- 1.8673E+02	- 1.8673E+02	8.84E+01	5.9985E-08	2.06493E-11	1.44E-14
f17	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
f18	0.0000E+00	6.9906E-153	0.0000E+00	0.0000E+00	1.4788E-152	0.0000E+00
f19	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
f20	- 7.5562E+03	- 2.2662E+03	- 5.50E+99	1.9273E+02	4.5626E+02	8.78E+100
f21	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
f22	3.0000E+00	3.0000E+00	3.0000E+00	0.0000E+00	0.0000E+00	2.52E-15
f23	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
f24	0.0000E+00	0.0000E+00	2.90E-13	0.0000E+00	0.0000E+00	1.61E-12
f25	1.7490E-03	2.8400E-02	2.48E-05	0.0000E+00	1.3179E-02	3.02E-05
f27	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
f28	- 1.2812E+03	- 2.7507E+07	- 1.06E+01	4.0402E+02	5.6521E+07	4.83E+00
f29	- 3.7912E-03	- 3.7912E-03	- 3.79E-03	1.7644E-18	0.0000E+00	3.09E-06
f30	0.00E+00	1.1527E-06	0.00E+00	0.00E+00	9.4711E-07	0.00E+00

Bold values are represent MBPFA algorithm can get high-precision results

Table 3 Mean and SD for comparing MBFPA with MFPA and BPFPA on 30 benchmark functions

	Mean			SD		
	MBPFA	MFPA	BPFPA	MBPFA	MFPA	BPFPA
<i>f</i> ₁	0.00E+00	321.9284	39.5022	0.00E+00	170.0287	20.5743
<i>f</i> ₂	0.00E+00	0.00E+00	4.8648e−23	0.00E+00	0.00E+00	2.089e−22
<i>f</i> ₃	0.00E+00	236,477,826.7226	40,308,138.5658	0.00E+00	146,754,099.6033	17,648,691.5919
<i>f</i> ₄	0.00E+00	807.3667	51.4	0.00E+00	342.6953	17.9417
<i>f</i> ₅	5.7759E−06	0.15245	0.10408	4.1738E−05	0.08634	0.033142
<i>f</i> ₆	0.00E+00	0.0000E+00	0.00E+00	0.00E+00	0.0000E+00	0.00E+00
<i>f</i> ₇	8.8818E−16	8.7275	4.0499	0.0000E+00	8.7275	0.86288
<i>f</i> ₈	0.0000E+00	3.2972	1.4447	0.0000E+00	0.88478	0.18451
<i>f</i> ₉	1.35E−31	1.35E−31	1.35E−31	6.6809E−47	6.6809e−47	6.6809e−47
<i>f</i> ₁₀	−6.08E+00	−8.6819	−7.8253	5.1538E−01	−8.6819	0.39279
<i>f</i> ₁₁	0.0000E+00	54.548	137.7442	0.0000E+00	10.3149	12.1861
<i>f</i> ₁₂	0.0000E+00	3.8028	7.3252	0.0000E+00	1.4868	4.7557
<i>f</i> ₁₃	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
<i>f</i> ₁₄	2.0681E+01	421.4537	234.5188	1.5593E+00	185.6032	70.9594
<i>f</i> ₁₅	−1.000E+00	−1.0000E+00	−1.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
<i>f</i> ₁₆	−1.8673E+02	−1.8673E+02	−1.8673E+02	5.9985E−08	0.00034889	0.010649
<i>f</i> ₁₇	0.0000E+00	9.4092	20.5805	0.0000E+00	3.2795	6.6107
<i>f</i> ₁₈	0.0000E+00	1.4077	1.3897	0.0000E+00	0.22267	0.28729
<i>f</i> ₁₉	0.0000E+00	7.6106	2.6414	0.0000E+00	2.0783	0.71014
<i>f</i> ₂₀	−7.5562E+03	−7493.523	−6733.0706	1.9273E+02	421.5054	339.4945
<i>f</i> ₂₁	0.0000E+00	0.0000E+00	2.8883e−28	0.0000E+00	0.0000E+00	6.459e−28
<i>f</i> ₂₂	3.0000E+00	3.0000E+00	3.0000E+00	0.0000E+00	1.8568e−15	1.0066e−15
<i>f</i> ₂₃	0.0000E+00	8.584e−130	2.4477e−26	0.0000E+00	2.2369e−129	8.8267e−26
<i>f</i> ₂₄	0.0000E+00	12.1634	11.8327	0.0000E+00	7.8409	6.5558
<i>f</i> ₂₅	1.7490E−03	1.4408E−04	0.014981	0.0000E+00	0.00024295	0.012595
<i>f</i> ₂₇	0.0000E+00	5.7103	0.0000E+00	0.0000E+00	22.072	2.2377
<i>f</i> ₂₈	−1.2812E+03	15,102.6218	−1.06E+01	4.0402E+02	4860.4391	8668.6782
<i>f</i> ₂₉	−3.7912E−03	−3.7912E−03	−3.7912E−03	1.7644E−18	1.7644e−18	1.7644e−18
<i>f</i> ₃₀	0.00E+00	0.00E+00	8.3804e−19	0.00E+00	0.00E+00	2.659e−18

Bold values are represent MBPFA algorithm can get high-precision results

Table 4 Parameter settings for seven algorithms

Algorithm	Parameter values
DE [15]	Scaling factor=0.5, Crossover probability=0.2
FPA [7]	The proximity probability p=0.8
PSO [3]	The cognitive parameter $c_1 = 1.5c_2 = 2.0$, inertia weight $\omega = 1$
GSA [21]	For GSA, the parameters are set to $G_0 = 100 \alpha = 20$
SOS [12]	Benefit factors randomly generated as either 1 or 2
TLBO [30]	Teaching factor T randomly generated as either 1 or 2
MBFPA	Sensory modality $c = 0.01$, power exponent a is increased from 0.1 to 0.3, proximity probability using Eq. (10)

on the other hand, because symbiosis is carried out between two individuals, the improvement in any one in the symbiosis stage will increase the convergence speed in the whole iterative process.

Although MBFPA is effective, it has some weaknesses. Compared with the original BOA and FPA algorithm, it is obvious that the algorithm becomes more complex, which will increase the running time of the algorithm; in addition,

Table 5 Statistical results of the 6 basic metaheuristic on the 49 problems

Algorithms	Better	Equal	Less than
DE	25	15	9
FPA	39	8	2
PSO	40	5	4
GSA	44	3	2
SOS	22	18	9
TLBO	25	18	6

*Where “Better,” “Equal,” and “less than” represent the number of problems that the performance of MBFPA is significantly better than, almost the same as, and significantly worse than the corresponding algorithm, respectively

increasing the mutualism stage is more dangerous. Because of the mutualism of each individual, it may lead to the decline of the individual who has been in the local optimal position. Therefore, how to selectively carry out mutually beneficial symbiosis of individuals is also a place worthy of consideration. The advantages and disadvantages of the above MBFPA algorithm are listed in Table 14.

6 Engineering design problems

In order to verify the performance of the proposed algorithm in constrained optimization problems, five engineering optimization problems (three-bar truss design, multi-plate disc clutch brake, pressure vessel, welded beam and speed reducer) from the structural field are compared and the best results are compared with other algorithms. In order to obtain a better solution, different maximum numbers of iterations and different population numbers are used for each different problem according to the number of optimization variables and constraints in all engineering optimization problems. The specific characteristics of these problems are given in Table 15.

6.1 Three-bar truss design problem

Three-bar truss design problem (see “Appendix A.1”) is the most common constraint optimization test problem, which is a minimization problem. The goal is to minimize the weight of three-bar truss, where the constraints are stress, deflection and buckling. Figure 24 shows the shape of the truss and the associated forces on the structure.

In this problem, the fixed population number is 30 and the maximum number of iterations is 100 and the optimal fitness value obtained by the proposed algorithm after 30 independent runs is $f(X) = 263.895843376$, $X = [0.788675132828, 0.408248295461]$. The best results of the proposed algorithm

will be compared with the optimal solution obtained in the previous literature in Table 16. The results show that the proposed algorithm can get a better solution in this problem and show good performance.

6.2 Multi-plate disc clutch brake

This problem requires the minimization of the weight of a multiple disc clutch brake (see “Appendix A.2”) by considering five discrete design variables: inner radius (r_1), outer radius (r_0), thickness of discs (t), actuating force (F), and number of friction surfaces (Z) [48]. Figure 25 shows a multiple disc clutch brake. The difficulty of the optimization problem increases, because the problem includes eight different constraints, which will result in only 70% of the feasible region in the solution space.

The fixed population number is 30 and the maximum number of iterations is 500, and the optimal fitness value obtained by the proposed algorithm after 30 independent runs is $f(X) = 0.235242457900804$, $X = [70, 90, 1, 600, 2]$. The results obtained by MBPFPA were compared with the optimal results of HHO [11], TLBO [30], WCA [47] and PVS [48]. As shown in Table 17, we can see that the proposed algorithm obtained the best solution in solving the problem.

6.3 Pressure vessel design problem

As a classical constrained engineering optimization problem, the goal of optimization is to minimize the total cost of welding, materials and manufacturing. The manufacturing configuration of the problem is shown in Fig. 26. The problem consists of four optimization variables: T_s (thickness of the shell), T_h (thickness of the head), r (inner radius), L (length of section without the head). At the same time, it contains four different constraints. The details are given in “Appendix A.3.”

The fixed population number is 20 and the maximum number of iterations is 200, and the optimal fitness value obtained by the proposed algorithm after 30 independent runs is $f(X) = 5885.3327736$, $X = [0.778168641371, 0.384649162626, 40.3196187241, 200.0]$. The obtained optimal solution is compared with other algorithms. Table 18 shows that the proposed algorithm is the best to deal with this problem and can obtain the optimal result.

6.4 Welded beam design problem

This problem requires designing the most appropriate height of weld ($\times 1$), length of weld ($\times 2$), height of beam ($\times 3$), and width of beam ($\times 4$) to minimize welding costs. Constraints include: shear stress, bending stress, buckling load

Table 6 Comparative results of the proposed algorithm with DE, FPA, PSO, GSA, SOS and TLBO

No		MBFPA	DE	FPA	PSO	GSA	SOS	TLBO
f_1	Mean	0	2.47E-04	1.92E+03	8.72E-07	3.40E-03	1.56E-135	1.43E-89
	SD	0	8.31E-05	5.26E+02	3.04E-06	1.86E-02	3.17E-135	2.13E-89
f_2	Mean	0	0	2.00E-12	5.08E-02	1.07E-18	0	0
	SD	0	0	4.76E-09	1.93E-01	1.24E-18	0	0
f_3	Mean	0	1.54E+02	1.65E+09	4.60E+01	2.10E+02	1.09E-129	1.94E83
	SD	0	7.20E+02	4.4E+08	5.52E+02	3.10E+02	2.24E-129	3.22E-83
f_4	Mean	0	0	1.85E+02	9.20E+00	1.42E+01	0	0
	SD	0	0	5.05E+02	7.84E+00	2.90E+01	0	0
f_5	Mean	1.42E-04	5.31E-02	3.08E-01	2.26E-02	3.99E-01	5.64E-04	1.19E-03
	SD	1.23E-04	1.43E-02	1.07E-01	7.62E-03	6.81E-01	2.99E-04	6.32E-04
f_6	Mean	0	0	1.14E-08	0	0	0	0
	SD	0	0	1.65E-08	0	0	0	0
f_7	Mean	8.88E-16	4.13E-03	7.35E+00	1.49E+00	8.58E-02	4.09E-15	6.45E-15
	SD	0	9.19E-04	1.22E+00	6.60E-01	3.39E-01	1.08E-15	1.79E-15
f_8	Mean	0	3.82E-03	1.75E+01	4.24E-02	2.53E+01	0	0
	SD	0	5.01E-03	4.75E+00	4.15E-02	5.86E+00	0	0
f_9	Mean	1.35E-31	1.35E-31	9.60E-10	1.35E-31	1.32E-18	1.35E-31	1.35E-31
	SD	6.68E-31	6.68E-47	1.90E-09	1.35E-47	1.44E-18	6.68E-47	6.68E-47
f_{10}	Mean	-4.9975	-9.1123	-4.8717	-7.7869	-8.3083	-9.0103	-4.654
	SD	0.47106	0.22017	0.48003	1.1805	0.78902	0.30491	0.47188
f_{11}	Mean	0	8.14E+01	1.86E+02	4.97E+01	3.66E+01	0	1.30E+01
	SD	0	9.60E+00	1.25E+01	1.62E+00	8.24E+00	0	5.20E+00
f_{12}	Mean	0	3.74E-02	2.11E+01	7.99E-03	2.62E-03	6.72E-70	1.19E-45
	SD	0	3.29E-02	1.94E+00	9.87E-03	2.71E-03	8.81E-70	9.83E-46
f_{13}	Mean	0	0	0	2.51E-05	1.39E-02	0	0
	SD	0	0	0	1.37E-04	4.24E-03	0	0
f_{14}	Mean	2.54E+01	6.62E+01	5.19E+03	3.53E+01	5.53E+01	2.81E+01	2.54E+01
	SD	9.20E-01	3.34E+01	2.32E+03	2.21E+01	6.43E+01	1.26E+00	6.71E-01
f_{15}	Mean	-1	-1	-1	-1	8.33E-01	-1	-1
	SD	0	0	2.66E-06	0	3.79E-01	0	0
f_{16}	Mean	-186.7309	-186.7309	186.7290	186.7290	-76.2892	-186.7305	-186.7305
	SD	3.16E-05	3.77E-14	1.88E-03	4.255E-14	67.2174	1.24E-03	1.95E-20
f_{17}	Mean	0	3.16E+02	1.54E+01	8.88E-01	4.09E+01	2.78E-46	2.62E-20
	SD	0	4.44E+01	4.77E+00	8.51E-01	1.43E+01	6.37E-46	7.50E-20
f_{18}	Mean	0	1.18E+00	2.34E+00	2.18E-01	2.61E+00	9.87E-56	1.42E-37
	SD	0	2.01E-01	4.02E-01	6.84E-02	2.15E+00	1.56E-55	1.19E-37
f_{19}	Mean	0	1.97E-03	3.20E+01	5.36E-02	5.50E-01	2.79E-69	4.84E-45
	SD	0	3.60E-03	7.21E+00	8.90E-02	1.15E+00	3.68E-69	5.80E-45
f_{20}	Mean	-6748.059	-9.55E+03	-6.37E+03	-6.85E+03	-2.51E+02	-1.04E+4	-8.12E+03
	SD	2.56E+02	5.66E+02	2.44E+02	7.98E+02	1.87E+02	6.51E+02	9.77E+02
f_{21}	Mean	0	0	4.83E-12	0	9.79E-19	0	0
	SD	0	0	1.28E-11	0	1.08E-18	0	0
f_{22}	Mean	3	3	3	3	3	3	3
	SD	1.375E-15	2.11E-15	1.52E-09	1.90E-15	9.34E-15	7.28E-16	5.53E-133
f_{23}	Mean	0	1.14E-39	3.00E-14	4.37E-127	3.54E-20	2.90E-204	1.51E-133
	SD	0	2.98E-39	5.37E-14	2.39E-126	3.42E-20	0.00E+00	5.93E-133
f_{24}	Mean	0	1.37E+02	7.50E-01	7.65E-03	3.58E+01	2.26E-18	6.64E-07
	SD	0	5.58E+01	2.75E+01	1.23E-02	2.85E+01	1.23E-17	1.73E-06
f_{25}	Mean	1.55E-03	2.63E-02	1.56E-02	2.55E-03	1.73E+00	7.61E-04	1.77E-03
	SD	1.61E-03	2.21E-02	1.29E-02	4.49E-03	3.55E+00	1.45E-03	2.76E-03

Table 6 (continued)

No		MBFPA	DE	FPA	PSO	GSA	SOS	TLBO
f26	Mean	-9.60E+00	-9.72E+00	-10.14E+00	-5.71E+00	-4.72E+00	-8.79E+00	-9.85E+00
	SD	1.33E+00	1.62E+00	3.21E-02	3.13E+00	1.96E+00	2.29E+00	1.09E+00
f27	Mean	0	2.94E-05	2.58E+02	1.78E-06	5.24E-02	3.99E-136	3.00E-90
	SD	0	1.03E-05	7.67E+01	9.07E-06	2.76E-01	1.36E-135	6.51E-90
f28	Mean	-824.5121	-722.8702	-319.1363	-1389.999	-376.832	-1389.9717	-620.1225
	SD	203.5617	210.6547	476.8181	0.0010726	48.446	0.13842	76.9352
f29	Mean	-3.7912E-03	-3.7912E-03	-3.7912E-03	-3.7912E-03	-3.7912E-03	-3.7912E-03	-3.7912E-03
	SD	1.76E-18	1.76E-18	3.73E-11	1.76E-18	1.43E-03	1.76E-18	1.76E-18
f30	Mean	4.77E-28	3.04E-05	6.77E-10	1.96E-13	3.46E+00	4.66E-24	3.00E-18
	SD	3.61E-30	1.67E-04	1.19E-09	1.04E-12	8.95E+00	9.02E-17	1.30E-17
f31	Mean	0	0	0	0.16667	0.00E+00	0.00E+00	0
	SD	0	0	0	0.46113	0.00E+00	0.00E+00	0
f32	Mean	1.9229e-09	0.014151	0.84197	0.013523	2.2912	1.7713e-06	6.005e-05
	SD	8.5913e-09	0.0266	0.59836	0.028011	2.5911	2.7075e-06	0.0002167
f33	Mean	0	0.80091	0.44679	1.5895e-23	32.2198	9.6583e-97	4.1953e-52
	SD	0	0.57182	0.20611	8.2719e-23	23.0425	3.0788e-96	1.3486e-51
f34	Mean	0.66667	1.1345	2106.5512	0.80044	1.4626	0.66667	0.66667
	SD	5.062e-09	0.52524	1082.586	0.45797	1.288	4.6695e-16	6.9636e-11
f35	Mean	0.998	0.998	0.99804	4.2063	13.8477	0.998	0.998
	SD	1.0909e-16	0	0.00011513	3.5086	6.5919	0	0
f36	Mean	0.39789	0.39789	0.39789	0.39789	0.39789	0.39789	0.39789
	SD	0	0	1.0909e-06	0	0	0	0
f37	Mean	-1.0316	-1.0316	-1.0316	-1.0316	-1.0316	-1.0316	-1.0316
	SD	6.4539e-16	6.7752e-16	7.5631e-08	6.3877e-16	4.3441e-16	6.7752e-16	6.7122e-16
f38	Mean	0.00030749	0.00068446	0.00070785	0.0012646	0.013252	0.00034321	0.00040824
	SD	1.0001e-10	0.00012951	0.00018082	0.0036397	0.0076868	0.0001677	0.00021674
f39	Mean	0.029788	0.12031	0.77598	0.071187	22.8601	0.017961	0.00021674
	SD	0.086304	0.12312	0.54825	0.12438	43.7677	0.037765	0.11913
f40	Mean	-3.8628	-3.8628	-3.8628	-3.8112	-0.68659	-3.8628	-3.8628
	SD	2.6543e-15	2.7101e-15	2.53e-08	0.19612	0.96964	2.7101e-15	2.7101e-15
f41	Mean	-3.3018	-3.3224	-3.2928	-3.2826	-0.69676	-3.2745	-3.2916
	SD	0.044744	3.5999e-09	0.024118	0.057155	0.43278	0.059277	0.051482
f42	Mean	0.00019444	0.0001291	39.1718	0.20049	0.72433	0.0034556	4.3409e-04
	SD	0.00045217	9.7444e-05	8.3372	0.53152	0.93427	0.018927	1.1576e-06
f43	Mean	1.5041	1.9565e-05	98.6619	0.03867	3.1383	0.099525	0.10525
	SD	1.1085	8.9344e-06	23.9594	0.083781	8.2229	0.084214	0.1112
f44	Mean	-1.0809	-1.0809	-1.0809	-1.0539	-0.83968	-1.0809	-1.0809
	SD	4.5168e-16	4.5168e-16	5.011e-10	0.05505	0.42543	4.5168e-16	4.5168e-16
f45	Mean	-1.4128	-1.3754	-1.0729	-1.2147	-0.079911	-1.3724	-1.1599
	SD	0.26942	0.10787	0.22864	0.29192	0.11968	0.23541	0.30714
f46	Mean	-0.59898	-0.53705	-0.40214	-0.54436	-0.0008114	-0.72055	-0.5387
	SD	0.28994	0.17137	0.18212	0.25042	0.0023572	0.40612	0.31296
f47	Mean	8.3234e-23	0	7.6125e-05	47.0197	94.0393	0	0
	SD	3.2682e-22	0	9.1474e-05	178.9391	243.8528	0	0
f48	Mean	49.3012	31.6276	67.9252	1166.8096	188.3431	0.0026592	38.8046
	SD	89.2502	70.2243	68.5145	1179.2897	86.1584	0.012047	80.7725
f49	Mean	1841.1397	306.1898	8988.2742	3453.4229	951.4124	229.5545	963.2594
	SD	2051.3933	104.6094	3948.2518	6351.4667	1589.0465	762.5573	1369.8109
The number of theoretical values		27	16	8	6	4	15	15

Bold numbers represent the best values

Table 7 Results of the p value benchmark functions

No	MBFPA VS DE	MBFPA VS FPA	MBFPA VS PSO	MBFPA VS GSA	MBFPA VS SOS	MBFPA VS TLBO
1	1.21178E-12	1.21178E-12	1.2117E-12	1.2117E-12	1.2117E-12	1.2117E-12
2	0.3337	1.7203E-12	0.0483	1.720E-12	0.3337	0.3337
3	1.2117E-12	1.2117E-12	1.2117E-12	1.2117E-12	1.2117E-12	1.2117E-12
4	1.00	1.2118E-12	1.1766E-12	5.7442E-10	1.00	1.00
5	3.0199E-11	3.0199E-11	3.0199E-11	3.0199E-11	8.1465E-05	1.8567E-09
6	1.00	1.2118E-12	1.00	1.00	1.00	1.00
7	1.2118E-12	1.2118E-12	1.2118E-12	1.2118E-12	3.9410E-12	4.4551E-13
8	1.2118E-12	1.2118E-12	1.2118E-12	1.2118E-12	1.00	1.00
9	1.00	1.2118E-12	1.00	1.2118E-12	1.2118E-12	1.00
10	2.8574E-11	3.0199E-11	6.2027E-04	3.5638E-04	8.1527E-11	2.0283E-07
11	1.2117E-12	1.2118E-12	1.211E-12	1.210E-12	1.148E-12	1.2117E-12
12	1.2117E-12	1.211E-12	1.211E-12	1.211E-12	1.00	1.2117E-12
13	1.00	1.00	0.0337	1.211E-12	1.00	1.00
14	3.019E-11	3.019E-11	0.0575	3.019E-11	3.689E-11	5.264E-04
15	1.00	1.2118E-12	1.00	0.0214	1.211E-12	1.00
16	9.1589E-11	6.6955E-11	1.260E-10	2.672E-04	0.4551	2.8210E-10
17	1.2117E-12	1.2117E-12	1.2117E-12	1.2117E-12	1.2117E-12	1.2117E-12
18	1.21178E-12	1.21178E-12	1.2117E-12	1.2117E-12	1.2117E-12	1.2117E-12
19	1.21178E-12	1.2117E-12	1.2117E-12	1.2117E-12	1.2117E-12	1.2117E-12
20	2.1544E-10	2.1544E-10	2.1544E-10	2.1544E-10	2.1544E-10	2.1544E-10
21	1.00	1.2118E-12	1.00	1.2118E-12	1.00	1.00
22	5.8157E-04	1.9242E-11	0.6578	1.9142E-11	0.0119	0.0812
23	1.2117E-12	1.2117E-12	1.2117E-12	1.2117E-12	1.2117E-12	1.2117E-12
24	1.2117E-12	1.2117E-12	1.2117E-12	1.2117E-12	1.2117E-12	1.2117E-12
25	4.9752E-11	4.9752E-11	4.9752E-11	4.9752E-11	4.9752E-11	4.9752E-11
26	4.9752E-11	1.8790E-09	0.0034	5.8721E-11	5.1859E-04	0.8135
27	1.2117E-12	1.2117E-12	1.2117E-12	1.2117E-12	1.2117E-12	1.2117E-12
28	3.0199E-11	3.0199E-11	0.0051	3.0199E-11	8.4848E-09	3.3679E-04
29	0.049	6.3188E-12	8.4226E-08	6.2647E-12	0.0109	0.0109
30	1.4051E-09	3.0047E-11	3.0047E-11	3.0047E-11	3.0047E-11	3.0047E-11
31	1.00	1.00	0.041865	1	1	1
32	3.01986E-11	3.01986E-11	3.01986E-11	3.01986E-11	5.49405E-11	4.97517E-11
33	1.21178E-12	1.21178E-12	1.21178E-12	1.21178E-12	1.21178E-12	1.21178E-12
34	3.01986E-11	3.01986E-11	2.37682E-07	0.00039881	2.7495E-11	3.4742E-10
35	0.005466034	7.57407E-12	1.04283E-0	7.49962E-12	0.005466034	0.005466034
36	1	1.21178E-12	1	1	1	1
37	0.021419179	5.14363E-12	0.749103252	1.44603E-10	0.021419179	0.090513169
38	3.0123E-11	3.0123E-11	3.0123E-11	3.0123E-11	0.001472867	0.011458939
39	2.49131E-06	3.15889E-10	0.211561237	7.11859E-09	0.002499392	0.332854692
40	0.04177393	4.08059E-12	0.468729621	6.31718E-13	0.04177393	0.04177393
41	0.447787971	1.87936E-05	0.19301787	2.54317E-11	0.371541283	0.981521904
42	0.096262831	3.01986E-11	0.077271976	1.72941E-07	5.57265E-10	3.01986E-11
43	3.01986E-11	3.01986E-11	5.49405E-11	0.005569939	9.88918E-11	1.95678E-10
44	1	1.21178E-12	0.000142057	2.62034E-13	1	1
45	0.039150716	1.48881E-06	0.376236712	3.00852E-11	4.77755E-05	0.662002664
46	0.958731491	7.59915E-07	0.970512775	3.01986E-11	0.662717173	0.047868861
47	0.000313354	1.44305E-11	0.01129043	1.4392E-11	0.000313354	0.000313354
48	0.428963389	0.00185748	0.001029216	2.8735E-06	4.76558E-06	0.841779758
49	0.00069125	1.28704E-09	0.952841775	0.04841038	1.52917E-05	0.043583548

Bold numbers represent the $p > 0.05$

Table 8 Friedman rank test for the mean performances obtained (for $f1-f49$ functions)

Algorithms	Mean rank	Rank
MBFPA	2.42	1
DE	3.93	4
FPA	5.56	7
PSO	4.52	5
GSA	5.82	6
SOS	2.61	2
TLBO	3.14	3

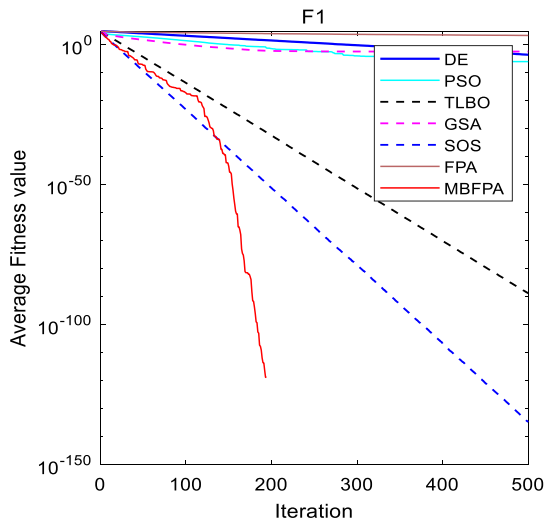


Fig. 2 Convergence graph for Sphere function

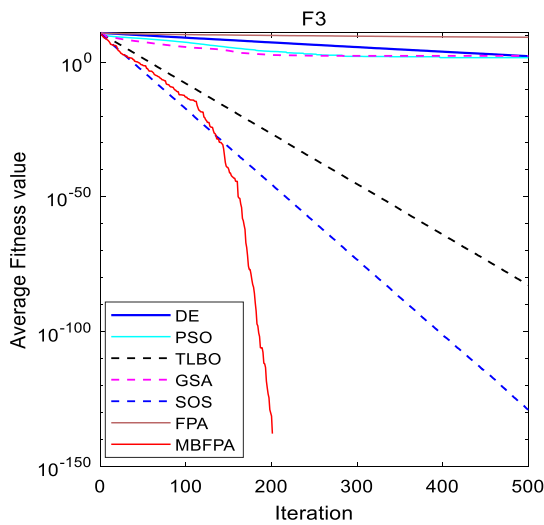


Fig. 3 Convergence graph for Cigar function

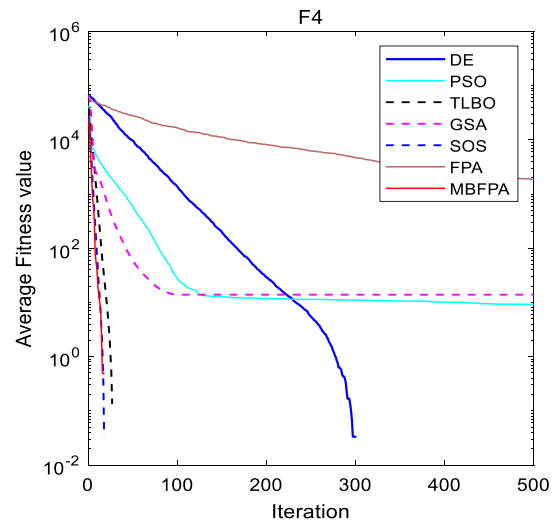


Fig. 4 Convergence graph for Step function

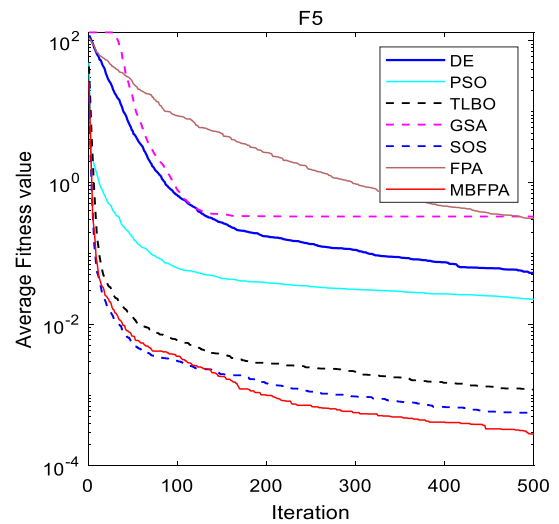


Fig. 5 Convergence graph for Quadraic function

on the bar, end deflection and side constraints [48]. The schematic diagram of the problem is shown in Fig. 27; the details are given in “Appendix A.4.” In this example, the fixed population number is 30 and the maximum number of iterations is 500, and the optimal fitness value obtained by the proposed algorithm after 30 independent runs is $f(X) = 1.72485185$, $X = [0.205730, 3.470473, 9.036623, 0.205729]$. The obtained optimal solution is compared with other 8 algorithms. In Table 19, we can conclude that the proposed algorithm can find the most suitable parameters and get the minimum fitness value.

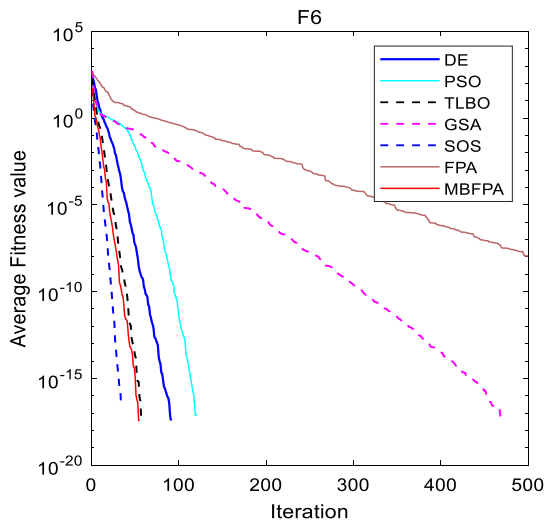


Fig. 6 Convergence graph for Bohachevsky function

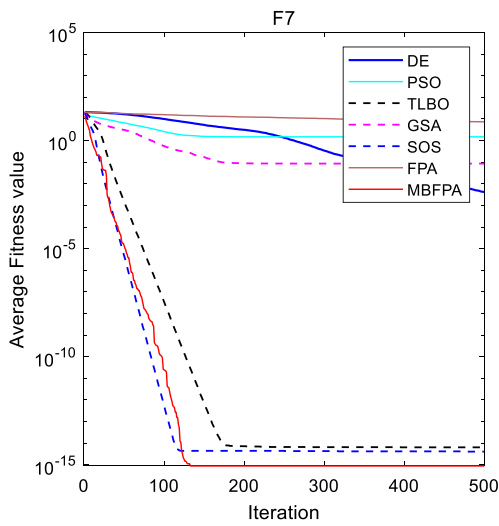


Fig. 7 Convergence graph for Ackley function

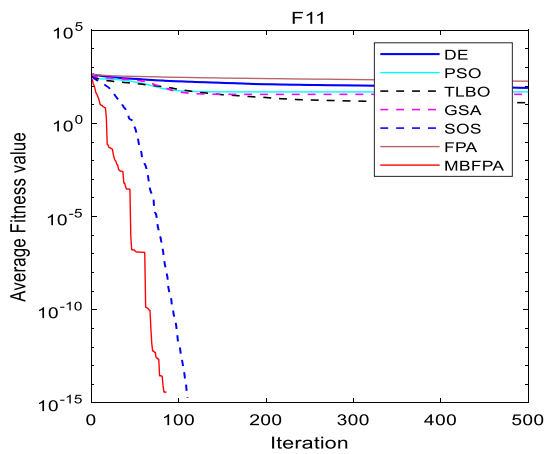


Fig. 8 Convergence graph for Rastrigin function

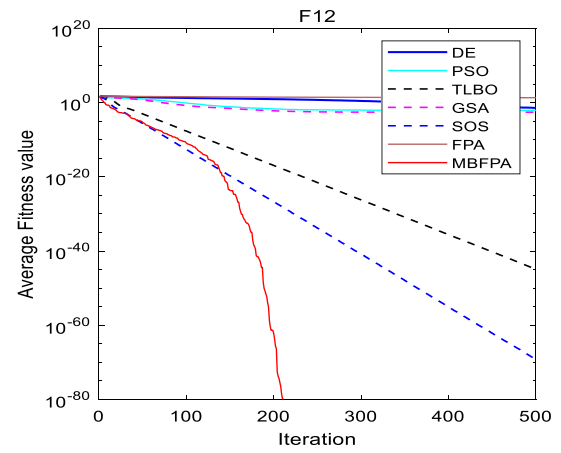


Fig. 9 Convergence graph for Alpine function

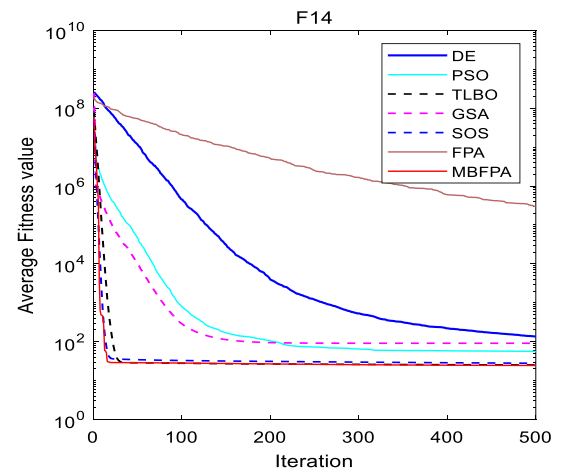


Fig. 10 Convergence graph for Rosenbrock function

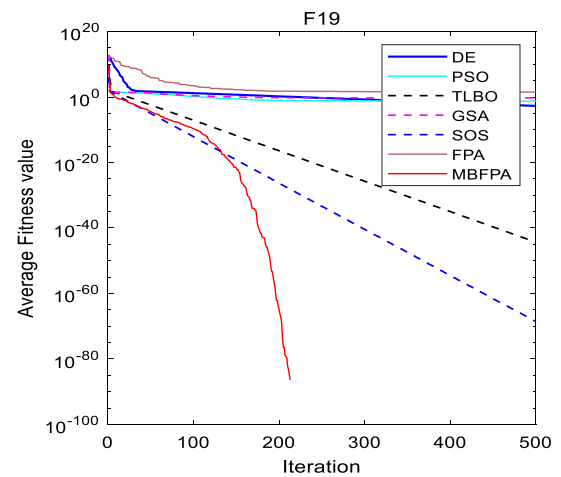


Fig. 11 Convergence graph for Schwefel 2.22 function

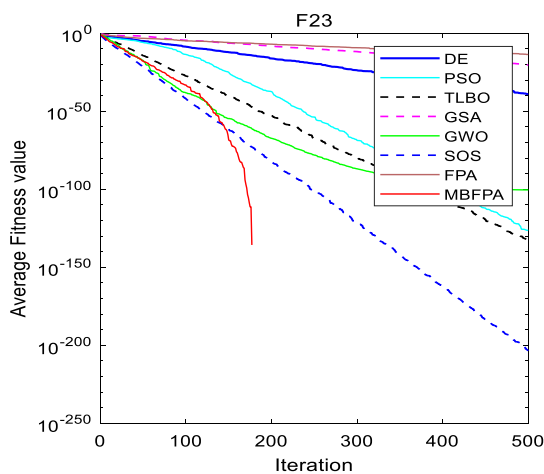


Fig. 12 Convergence graph for Matyas function

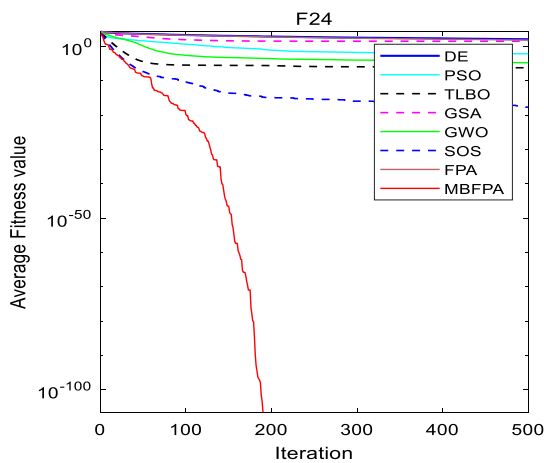


Fig. 13 Convergence graph for Powersum function

6.5 Speed reducer

Speed reducer (as shown in Fig. 28) is designed to minimize the weight of the reducer. There are mainly 7 variables to be

optimized: the face width($\times 1$), module of teeth($\times 2$), number of teeth on pinion($\times 3$), length of the first shaft between bearings($\times 4$), length of the second shaft between bearings($\times 5$), diameter of the first shaft($\times 6$) and diameter of the first shaft($\times 7$) [90]. In this problem, there are as many as 11 constraints, which is a challenging optimization problem, mainly to minimize the weight under the condition of bending stress of the gear teeth, surface stress, transverse deflections of the shafts and stresses in the shafts [90]. The details are given in “Appendix A.5.”

The optimal fitness value obtained by the proposed algorithm after 30 independent runs is $f(X) = 2994.341315$, $X = [3.5 \ 0.7 \ 17 \ 7.3 \ 7.7153199122 \ 3.35021466 \ 5.28665446]$. The optimal solutions obtained by this paper are compared with those obtained by the other five algorithms. Table 20 shows the detailed comparison. It can be concluded that the proposed algorithm has advantages in the optimization under complex constraints and can obtain better solutions.

7 Conclusions and future work

In recent years, butterfly optimization algorithm has been proposed as a new heuristic algorithm, which is widely used in global optimization problems. In BOA algorithm, individuals can find the most fragrant position by simulating the foraging process of butterflies and then find the optimal solution in the global optimization. The random number disturbance is added to the optimal position during the update of the butterfly optimization algorithm, which avoids individual crowding, but it brings a problem: it is difficult to make the most of the current the leading role of the optimal position leads to the possibility of missing the optimal position in actual application. This will cause the algorithm to have low accuracy and slow convergence. Therefore, this paper proposes a hybrid optimization algorithm for butterfly and flower pollination base on mutualism mechanism, which increases the speed and accuracy of algorithm exploitation through the mutualism mechanism. At the same time,

Table 9 Parameter settings for six algorithms

Algorithm	Parameter values
PSOGWO [82]	The parameter is a linearly decreased from 2 to 0, $c1 = c2 = c3 = 0.5, w = 0.5 + rand/2$
SCSCA [83]	Constant $a = 2, r1 = a \cdot r^*((a)/max_iter), r2 = 2 * pi * rand, r3 = r2, r4 = rand$
MFPA [53]	Switch probability $p = 0.8$, cloning array = [1, 1, 1, 1, 1-9], $r1 = 1, r2 = 3$
HFPSO [82]	The parameter $a = 0.2, B_0 = 2, r = 1, c1 = c2 = 1.49445, w_i = 0.9, w_f = 0.5$
MDE-WOA [84]	crossover probability $cr = 0.9$, scaling factor $F = 0.8$, neighborhood of radius $k = 3$, weight factor $\omega = 0.5$
MBFPA	sensory modality $c = 0.01$, power exponent a is increased from 0.1 to 0.3, proximity probability using Eq. (10)

Table 10 Statistical results of the five hybrid metaheuristics on the 49 problems

Algorithms	Better	Equal	Less than
PSOGWO	48	1	0
SCSCA	38	5	6
HFPSO	37	8	4
MFPA	24	14	11
MDE-WOA	38	5	6

*Where “Better,” “Equal” and “less than” represent the number of problems that the performance of MBFPA is significantly better than, almost the same as and significantly worse than the corresponding algorithm, respectively

because the flower pollination algorithm is mixed, the alienated pollination of flowers is propagated through Levy flight, which can increase the algorithm’s exploration capability and introduce a dynamic switching probability equilibrium exploration and exploitation process. The results show that the butterfly optimization algorithm and the flower pollination algorithm can get good results through the appropriate mechanism, and in terms of nature, butterflies and flowers are two inseparable creatures. Compared with other metaheuristics, this algorithm can find better or equal function values in most benchmark problems. From the results of the five engineering examples, the proposed algorithm has great advantages in solving constrained optimization problems and can get better solutions.

In the future research, in order to verify the ability of the hybrid algorithm to solve the constrained optimization problem, we will focus on the application of the proposed algorithm to the solution of practical problems. Since the vehicle routing problem (VRP) was proposed in 1959, it has been widely concerned by researchers because of its wide application and great economic value, such as postal delivery problem, vehicle scheduling problem and computer network topology problem [107]. VRP is a NP hard problem, which is difficult to solve with accurate algorithm. At present, metaheuristics have become prominent approaches in tackling complex and multi-objective problems [108]. So, metaheuristic algorithm is the main method to solve VRP. However, the core of using metaheuristic algorithm to solve VRP problem is how to encode the individuals according to the background of the problem. The quality of coding will directly affect the complexity of algorithm discretization and the performance of the algorithm. In addition, how to discretize the proposed algorithm so that the algorithm can play a good effect in the discrete problem space is also an important issue. Therefore, the design of reasonable coding and discrete version of MBPFA has become the key to VRP

path problems in the future. In addition, we have identified four areas that deserve further study in the future.

Hybrid metaheuristics combines the advantages of different metaheuristics to provide simpler, better and faster solutions for many complex problems [76]. A new metaheuristic algorithm is proposed, which actually increases the diversity of metaheuristic algorithm in the update strategy. Different metaheuristic algorithms let the population update according to a specific update strategy. However, how to combine these update strategies in an efficient and reasonable way will be an interesting and exciting thing. Therefore, by coupling the characteristics of different strategies, a more effective hybrid algorithm will be the focus of future algorithm research.

In the absence of gradient information, the metaheuristic algorithm is easier to implement than the precise search algorithm. In the actual performance test, the parameters of metaheuristic algorithm greatly affect the performance of the algorithm. Most of the given reasonable parameters are obtained through a large number of experiments. Therefore, it is necessary to develop a function with adaptive parameter adjustment. Or, we should pay more attention to the non-parametric metaheuristic algorithm, because it can increase our thinking on the algorithm itself, rather than focusing on the adjustment of algorithm parameters. However, developing parameterless metaheuristic algorithms is yet an open problem and needs to be thoroughly studied [77]. Therefore, we should pay more attention to the development of non-parametric meta heuristic algorithm in the future.

Artificial neural networks (ANNs) [109] are well-known techniques utilized in learning, approximating and investigating various classes of complex problems [110]. ANNs are used for prediction purposes in most cases, and it tries to close the gap among the predicted solution and the given target solution by frequently adjusting the values of weight functions [111]. There are two methods to adjust the weight: metaheuristic-based trainers and gradient-based trainers. Among them, the gradient-based trainer has better performance in local search [112], while the metaheuristic-based trainers has better performance in global search and can effectively avoid local optimization [110]. Some works use genetic algorithm (GA), differential evolution (DE) and evolutionary strategy (ES) to improve artificial neural network [113–122]. The results affirm the satisfactory results of these hybrid MLP structures [123]. Although a lot of work has been done so far, searching for global results of hybrid MLP networks is still an open question [124, 125]. From No Free Lunch theorem [126], a new superior SMHA can still be designed to be integrated with MLP networks [123]. Therefore, metaheuristic algorithm is still worthy of optimizing ANN in the future.

Table 11 Comparative results of the proposed algorithm with PSO, SCSCA, HFPSO, MFPA and MDE-WOA

No		MBFPA	PSOGWO	SCSCA	HFPSO	MFPA	MDE-WOA
f1	Mean	0	1464.2814	4.5172e-14	3.8625e-05	0.59529	2.5799e-81
	SD	0	7292.4107	8.1704e-14	2.6278e-05	2.0744	1.3518e-80
f2	Mean	0	0.025455	4.2455e-05	0.025402	0	0.15241
	SD	0	0.13912	7.2352e-05	0.13913	0	0.31004
f3	Mean	0	2,563,406,007.5963	1.5467e-07	1104.9195	2,318,419.1366	1.2469e-77
	SD	0	8,609,227,641.8811	4.8979e-07	1943.4194	11,248,886.7624	6.5763e-77
f4	Mean	0	2443.0333	0	0.63333	36.9333	0
	SD	0	7896.028	0	0.96431	52.8804	0
f5	Mean	1.42E-04	5.4452	0.0014015	0.018369	0.29065	0.0008511
	SD	1.23E-04	15.2685	0.0015904	0.0074209	0.18338	0.0013862
f6	Mean	0	3.3781e-07	1.014e-15	0	0	0
	SD	0	1.2539e-06	3.1393e-15	0	0	0
f7	Mean	8.88E-16	6.5056	3.0496e-08	0.55656	12.7309	2.5461e-15
	SD	0	7.3602	7.0176e-08	0.72773	5.9157	2.0298e-15
f8	Mean	0	50.2158	2.8462e-14	0.011595	0.31272	0.014832
	SD	0	89.8593	6.4502e-14	0.020654	0.2921	0.081237
f9	Mean	1.35E-31	0.0021402	0.00020701	2.453e-29	1.35E-31	0.0059975
	SD	6.68E-31	0.010311	0.00031822	7.2693e-29	6.6809e-47	0.020154
f10	Mean	-4.9975	-7.7958	-5.8327	-8.8304	-8.7811	-5.1218
	SD	0.47106	1.241	0.55664	0.56524	0.44998	0.66431
f11	Mean	0	111.6577	4.5854e-13	50.3121	69.7322	0
	SD	0	117.2388	2.0128e-12	18.0887	28.1146	0
f12	Mean	0	9.9146	2.0812e-08	0.015052	0.050841	2.2761e-56
	SD	0	14.6743	3.1042e-08	0.014775	0.16915	8.5285e-56
f13	Mean	0	0.42966	1.317	0.020304	0	0.082758
	SD	0	0.5562	2.0102	0.029206	0	0.11231
f14	Mean	2.54E+01	80,303.3203	0.02902	35.8411	131.0716	11.9217
	SD	9.20E-01	242,372.5379	0.046282	22.135	80.0088	13.9223
f15	Mean	-1	-0.99996	-0.99988	-1	-1	-0.99868
	SD	0	0.00013291	0.00012459	-1	-1	0.0016975
f16	Mean	-186.7309	-186.566	-186.1738	-186.7309	-186.7309	-186.7217
	SD	3.16E-05	0.62993	0.7096	1.9029e-14	1.669e-14	0.022078
f17	Mean	0	59.8604	4.4904e-09	1.0771	9.9161	50.9505
	SD	0	165.8967	1.5732e-08	0.56174	7.6294	96.5749
f18	Mean	0	1.3904	1.384e-09	0.1452	2.9532	2.3125e-11
	SD	0	2.4368	1.7288e-09	0.16558	0.5579	8.8309e-11
f19	Mean	0	194.7062	8.6921e-08	0.0050617	0.14495	9.8365e-55
	SD	0	3.60E-03	1.3219e-07	0.0013767	0.37695	4.0815e-54
f20	Mean	-6748.059	-6824.173	-12,569.4711	-7056.146	-8035.8558	-12,492.6458
	SD	2.56E+02	796.7108	0.026996	895.3158	757.2182	187.6248
f21	Mean	0	0.00055731	0.00037454	6.0032e-29	0	0.080391
	SD	0	0.0017441	0.00042647	2.8114e-28	0	0.35879
f22	Mean	3	3.0004	3.0025	3	3	8.4122
	SD	1.375E-15	0.0012608	0.0034966	3	1.3601e-15	11.0092
f23	Mean	0	3.4108e-05	7.1382e-21	2.4913e-29	9.4404e-67	1.8953e-193
	SD	0	0.0001115	3.3442e-20	1.0886e-28	5.1634e-66	0
f24	Mean	0	133.1876	9.9564e-15	0.038249	1.546	1.8865e-33
	SD	0	544.6846	2.0631e-14	0.022975	2.6536	1.0328e-32
f25	Mean	1.55E-03	0.14355	0.49844	0.0024393	9.4404e-67	3.8156
	SD	1.61E-03	0.28952	0.44027	0.003987	2.8728e-06	5.0596

Table 11 (continued)

No		MBFPA	PSOGWO	SCSCA	HFPSO	MFPA	MDE–WOA
f26	Mean	−9.60E+00	−7.1036	−10.1506	−5.7256	−9.6434	−8.8565
	SD	1.33E+00	3.5049	0.002894	3.5189	1.5555	2.14
f27	Mean	0.00E+00	292.9653	6.1751e−15	5.5318e−05	0.47711	2.7192e−82
	SD	0.00E+00	929.781	1.564e−14	3.7834e−05	2.0492	1.1081e−81
f28	Mean	−824.5121	147,400.8462	−1222.0321	−273.5306	5181.433	−1211.9538
	SD	203.5617	295,687.4565	345.1971	4990.0141	5535.9984	252.6967
f29	Mean	−0.0037912	−0.0037912	−0.0037912	−0.0037912	−0.0037912	−0.0037912
	SD	1.76E−18	4.2949e−08	4.0209e−08	1.5152e−18	1.7644e−18	1.8722e−07
f30	Mean	4.77E−28	1.2828e−05	5.5778e−06	5.6072e−13	0	0.00022238
	SD	3.61E−30	2.269e−05	7.27e−06	1.2943e−12	0	0.00047019
f31	Mean	0	0.8	0	0	0	0
	SD	0	1.5844	0	0	0	0
f32	Mean	1.9229e−09	1.8528	0.0064765	0.041491	4.6792e−26	1.0921
	SD	8.5913e−09	3.7306	0.0080634	0.10125	2.4723e−25	2.2718
f33	Mean	0	1.5746	4.966e−11	6.9216e−10	3.1215e−09	1.0772
	SD	0	3.6314	2.0759e−10	8.5116e−10	1.4231e−08	3.4025
f34	Mean	0.66667	51,445.1189	0.24509	1.0709	7.8827	0.43964
	SD	5.062e−09	139,833.8416	0.019825	0.88246	15.7031	0.21716
f35	Mean	0.998	4.2503	1.2297	2.4441	1.0974	1.8565
	SD	1.0909e−16	4.818	0.56408	2.3096	0.30331	1.3407
f36	Mean	0.39789	0.47789	0.3984	0.39789	0.39789	0.55809
	SD	0	0.43691	0.00090422	0	0	0.60703
f37	Mean	−1.0316	−1.0312	−1.0316	−1.0316	−1.0316	−1.0316
	SD	6.4539e−16	0.0019509	5.963e−05	5.3761e−16	6.7752e−16	1.3244e−09
f38	Mean	0.00030749	0.0078717	0.00054909	0.0011534	0.00044668	0.00081112
	SD	1.0001e−10	0.0096897	0.00036899	0.0036456	0.00032098	0.00060613
f39	Mean	0.029788	2.5086	4.0528	0.04551	0.010766	11.36
	SD	0.086304	6.0719	3.6243	0.092229	0.030062	13.1579
f40	Mean	−3.8628	−3.8617	−3.8314	−3.8628	−3.8628	−3.8063
	SD	2.6543e−15	0.0026512	0.054714	2.3134e−15	2.7101e−15	0.062821
f41	Mean	−3.3018	−3.1873	−3.1687	−3.2906	−3.2906	−3.0761
	SD	0.044744	0.12287	0.095615	0.053616	0.053616	0.16108
f42	Mean	0.00019444	49.2069	0.00030119	0.036728	10.1681	0.0025115
	SD	0.00045217	104.5129	0.00051898	0.074892	7.8628	0.0026091
f43	Mean	1.5041	169.8461	0.00083634	0.0062672	52.9936	0.038273
	SD	1.1085	397.1979	0.001669	0.023855	47.253	0.089644
f44	Mean	−1.0809	−1.0764	−1.0806	−1.0584	−1.0809	−1.0674
	SD	4.5168e−16	0.024704	0.00042078	0.051289	4.5168e−16	0.041287
f45	Mean	−1.3128	−1.0793	−0.70596	−1.1734	−0.93253	−0.4832
	SD	0.26942	0.39449	0.30924	0.32426	0.37825	0.26011
f46	Mean	−0.59898	−0.38605	−0.10057	−0.70214	0.37825	−0.024302
	SD	0.28994	0.33593	0.10439	0.34045	0.20139	0.049538
f47	Mean	8.3234e−23	0.33593	0.58226	23.5098	0	23.5098
	SD	3.2682e−22	178.933	0.75704	128.7687	0	128.7687
f48	Mean	49.3012	1423.275	2236.2919	1122.6836	20.7998	2517.5172
	SD	89.2502	4132.7971	4922.6359	1374.7564	56.1376	3962.3547
f49	Mean	1841.1397	6962.7044	41,083.4969	5948.7012	703.3272	50,101.3955
	SD	2051.3933	8749.0243	39,585.1198	12,301.3318	1030.9211	26,026.7566

Bold numbers represent the best values

Table 12 Results of the p-value benchmark functions

No	MBFPA VS PSOGWO	MBFPA VS SCCSA	MBFPA VS HFPSO	MBFPA VS MFPA	MBFPA VS MDE–WOA
1	1.21178E–12	1.21178E–12	1.2117E–12	1.2117E–12	1.21178E–12
2	1.7203E–12	1.7203E–12	2.69685E–12	0.333710696	1.72025E–12
3	1.21178E–12	1.21178E–12	1.2117E–12	1.2117E–12	1.21178E–12
4	5.83738E–09	1	6.13278E–05	1.20487E–12	1
5	3.0199E–11	3.83067E–05	3.0199E–11	3.0199E–11	0.239849991
6	0.005584312	0.000313354	1.00	1.00	1
7	1.2118E–12	1.2118E–12	1.2118E–12	1.2118E–12	5.80026E–05
8	1.2118E–12	1.30408E–07	1.2118E–12	1.2118E–12	0.333710696
9	1.21178E–12	1.2118E–12	1.69435E–08	1	1.21178E–12
10	0.051877131	3.01986E–11	0.000189162	0.001236185	3.01986E–11
11	1.2117E–12	0.021577192	1.211E–12	1.210E–12	1
12	1.2117E–12	1.211E–12	1.211E–12	1.211E–12	1.21178E–12
13	1.27173E–05	0.000145517	6.60964E–05	1	2.93429E–05
14	7.38908E–11	3.01986E–11	1.09069E–05	5.57265E–10	0.185766856
15	1.21178E–12	1.2118E–12	1.00	1	1.21178E–12
16	9.06321E–08	3.01986E–11	1.01242E–10	3.18271E–11	4.4205E–06
17	1.2117E–12	1.2117E–12	1.2117E–12	1.2117E–12	1.21178E–12
18	1.21178E–12	1.21178E–12	1.2117E–12	1.2117E–12	1.21178E–12
19	1.21178E–12	1.2117E–12	1.2117E–12	1.2117E–12	1.21178E–12
20	0.162375022	3.01986E–11	0.529782491	4.35308E–05	3.01797E–11
21	1.21178E–12	1.2118E–12	1.26686E–05	1	1.21178E–12
22	1.92421E–11	1.9242E–11	0.033375869	0.033941098	1.92421E–11
23	1.21178E–12	1.21178E–12	4.57359E–12	1.21178E–12	1.21178E–12
24	1.2117E–12	1.2117E–12	1.2117E–12	1.2117E–12	1.21178E–12
25	6.01039E–08	3.01986E–11	0.501143668	5.49405E–11	3.01986E–11
26	1.79266E–10	2.27167E–09	5.50767E–06	0.008235911	7.90834E–10
27	1.21178E–12	1.2117E–12	1.2117E–12	1.2117E–12	1.21178E–12
28	7.11859E–09	0.000140669	0.510597937	1.72941E–07	2.43271E–05
29	1.21178E–12	1.21178E–12	5.1893E–07	1	1.21178E–12
30	3.00663E–11	3.0047E–11	3.0047E–11	3.0047E–11	3.00663E–11
31	0.00557794	1	1	1	1
32	3.01986E–11	3.01986E–11	3.01986E–11	2.24487E–11	3.01986E–11
33	1.21178E–12	1.21178E–12	1.21178E–12	1.21178E–12	1.21178E–12
34	3.01986E–11	3.01986E–11	3.01986E–11	3.01986E–11	0.077271976
35	7.57407E–12	7.57407E–12	2.52423E–06	0.45517435	7.57407E–12
36	1.21178E–12	1.21178E–12	1	1	1.21178E–12
37	5.14363E–12	5.14363E–12	0.000101743	0.021419179	5.14363E–12
38	3.0123E–11	3.0123E–11	3.0123E–11	0.002035545	3.0123E–11
39	9.83289E–08	6.69552E–11	0.706171488	8.84109E–07	4.97517E–11
40	4.08059E–12	4.08059E–12	8.2176E–08	0.04177393	4.08059E–12
41	5.57217E–07	9.91325E–09	0.252190204	0.441038361	3.05844E–10
42	1.38525E–06	0.270705338	0.270705338	3.01986E–11	1.8731E–07
43	0.222572896	3.01986E–11	3.01986E–11	3.35195E–08	6.06576E–11
44	1.21178E–12	1.21178E–12	1.19448E–13	1	1.21178E–12
45	2.42892E–05	1.40656E–09	0.869757328	0.002242796	6.67125E–11
46	0.000421751	8.15274E–11	0.102326268	0.007286735	3.01986E–11
47	1.44305E–11	1.44305E–11	0.47840956	0.000313354	1.44305E–11
48	1.63506E–05	2.43863E–09	2.20308E–05	0.007617064	1.42942E–08
49	0.011710684	8.15274E–11	0.395267011	0.016954881	4.50432E–11

Bold numbers represent the $p > 0.05$

Table 13 Friedman rank test for the mean performances obtained (for $f1-f49$ functions)

Algorithms	Mean-rank	Rank
MBFPA	1.94	1
PSOGWO	5.13	4
SCCSA	3.38	7
HFPSO	3.41	5
MFPA	3.22	2
MDE-WOA	3.92	3

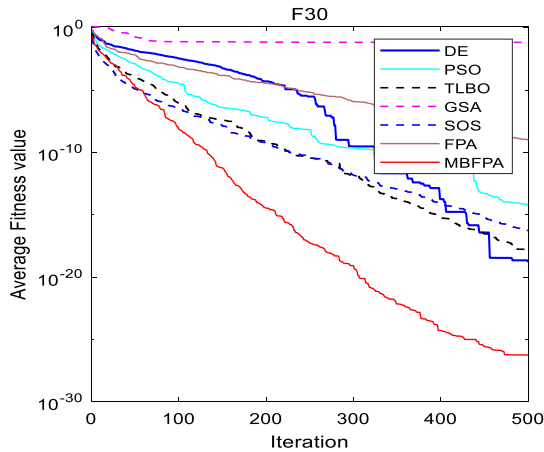


Fig. 14 Convergence graph for Leon function

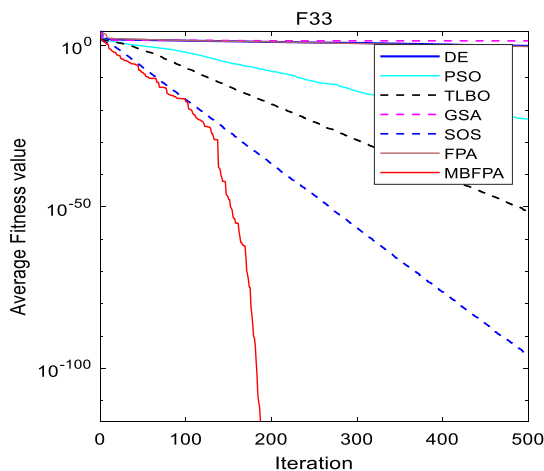


Fig. 15 Convergence graph for Zakharov function

In practical engineering applications, such as product design, wing design, investment allocation, urban planning and other issues, they have more than one single goal, and they need to meet a large number of constraints at the

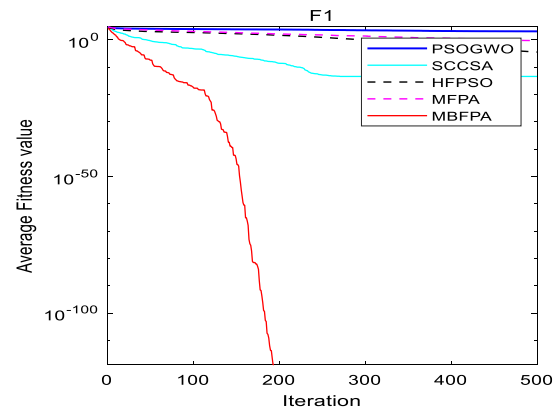


Fig. 16 Convergence graph for Sphere function

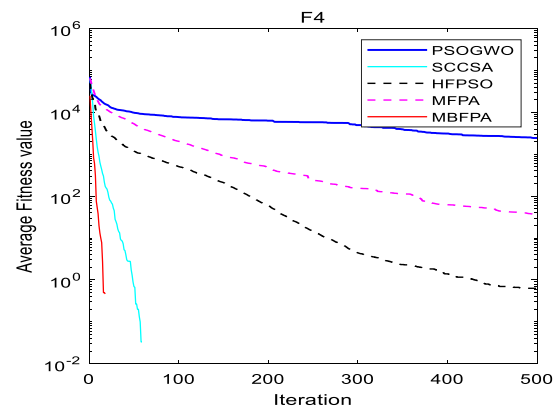


Fig. 17 Convergence graph for Step function

same time. In addition, these problems have many decision variables, which are regarded as large-scale optimization problems. Currently, most of the existing metaheuristic algorithms proposed either require thousands of expensive exact function evaluations to obtain acceptable solutions or focus on solving low-dimensional and expensive optimization problems [127]. This greatly limits the application space of metaheuristic algorithm and seriously affects the performance and efficiency of the algorithm. However, the rise of surrogate-assisted metaheuristic algorithms (SAEAs) offers possibilities to solve this problem. The main parts of SAEAs are the surrogates and the evolutionary optimizer. SAEAs use proxy surrogates to reduce the amount of computation brought by the fitness evaluation in the iterative process [128]. In the future development process of metaheuristic algorithms, surrogate-assisted metaheuristic algorithms will become more and more worthy of attention.

Table 14 Advantages and disadvantages of the MBFPA

Advantages	Disadvantages
The proposed algorithm has the advantages of BOA and FPA; Avoid premature convergence, strong global search capability, fast convergence speed and high optimization precision	Long running time The mutualism stage of sequential execution may lead to further declines in positions that are already locally optimal

Table 15 Brief description of engineering design features (F/S: the proportion of feasible solutions)

No	Problems name	Dim	Continuous design variables	Discrete design variables	Number of constraints	Active constraints	F/S	Objective
1	Three-bar truss	2	2	0	3	N/A	N/A	Minimize weight
2	Multi-plate disc clutch brake	5	0	5	8	1	0.7	Minimize weight
3	Pressure vessel	4	2	2	4	2	0.4	Minimize cost
4	Welded beam	4	4	0	7	2	0.035	Minimize cost
5	Speed reducer	7	6	1	11	3	0.004	Minimize weight

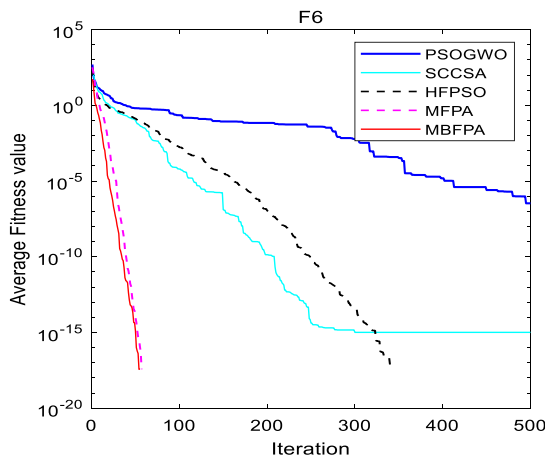


Fig. 18 Convergence graph for Bohachevsky function

Finally, we hope that this paper will inspire researchers in metaheuristics and optimization field.

Table 16 Comparison of results for three-bar truss design problem

Algorithm	x_1 (A1)	x_2 (A2)	Optimal weight
MBFPA	0.788675132828	0.408248295461	263.895843376
DEDS [85]	0.78867513	0.40824828	263.8958434
MVO [23]	0.78860276	0.408453070000000	263.8958499
GOA [86]	0.788897555578973	0.407619570115153	263.895881496069
MFO [87]	0.788244771	0.409466905784741	263.8959797
PSO–DE [88]	0.7886751	0.4082482	263.8958433
SSA [89]	0.788665414	0.408275784444547	263.8958434
MBA [90]	0.7885650	0.4085597	263.8958522
WCA [91]	0.788651	0.408316	263.895843

Bold values are represent MBPFPA algorithm can get high-precision results

Acknowledgements This work is supported by National Science Foundation of China under Grant No. 61563008, and by Project of Guangxi Natural Science Foundation under Grant No. 2018GXNSFAA138146.

Compliance with ethical standards

Conflict of interest The authors declare no conflict of interest.

Appendix A

A.1. Three-bar truss design

Minimize $f(x) = (2\sqrt{2}x_1 + x_2) \times l$.
 Subject to $g_1(x) = \sqrt{2}x_1 + x_2/\sqrt{2}x_1^2 + 2x_1x_2P - \sigma \leq 0$;
 $g_2(x) = x_2/\sqrt{2}x_1^2 + 2x_1x_2P - \sigma \leq 0$;
 $g_3(x) = 1/\sqrt{2}x_2 + x_1P - \sigma \leq 0$;

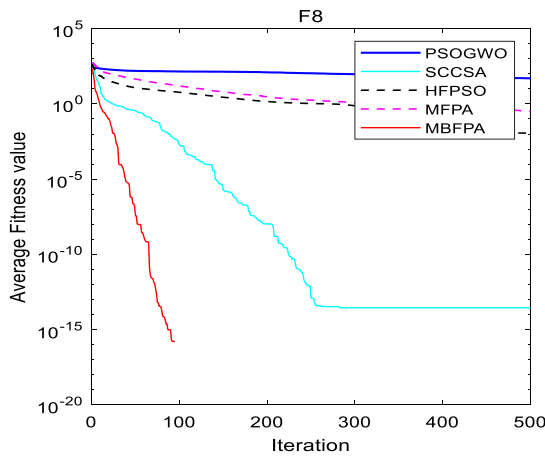


Fig. 19 Convergence graph for Griewank function

Variable range $0 \leq x_1, x_2 \leq 1$.
 where $l = 100 \text{ cm}$; $P = 2 \text{ kN/cm}^2$; $r = 2 \text{ kN/cm}^2$.

A.2. Multi-plate disc clutch brake design problem

Minimize $f(x) = \pi(x_2^2 - x_1^2)x_3(x_5 + 1)\rho$.

Subject to:

$$\begin{aligned}
 &g_1(x) = x_2 - x_1 - \Delta R \geq 0, \\
 &g_2(x) = L_{\max} - (x_5 + 1)(x_3 + \delta) \geq 0 \\
 &g_3(x) = P_{\max} - P_{rz} \geq 0, \quad g_4(x) = P_{\max} v_{sr \max} - P_{rz} v_{sr} \geq 0 \\
 &g_5(x) = v_{sr \max} - v_{sr} \geq 0, \quad g_6(x) = T_{\max} - T \geq 0 \\
 &g_7(x) = M_h - sM_s \geq 0, \quad g_8(x) = T \geq 0 \\
 &\text{where } M_h = \frac{2}{3} \mu x_4 x_5 \frac{x_2^3 - x_1^3}{x_2^2 - x_1^2}, \quad w = \frac{\pi n}{30} \text{ rad/s}, \\
 &A = \pi(x_2^2 - x_1^2) \text{ mm}^2, \\
 &P_{rz} = \frac{x_4}{A} \text{ N/mm}^2, \quad V_{sr} = \frac{\pi R_{sr} n}{30} \text{ mm/s}, \quad R_{sr} = \frac{2}{3} \frac{x_2^3 - x_1^3}{x_2^2 - x_1^2} \text{ mm} \\
 &T = \frac{I_z \pi n}{30(M_h + M_f)} \text{ mm}, \quad \Delta r = 20 \text{ mm}, \quad L_{\max} = 30 \text{ mm}, \quad \mu = 0.6 \\
 &T_{\max} = 15s, \quad \mu = 0.5, \quad s = 1.5, \quad M_s = 40 \text{ Nm}, \\
 &P_{\max} = 1 \text{ MPa}, \quad \rho = 0.0000078 \text{ kg/mm}^3, \\
 &v_{sr \max} = 10 \text{ m/s}, \quad \delta = 0.5 \text{ mm}, \quad s = 1.5 \\
 &T_{\max} = 15s, \quad n = 250 \text{ rpm}, \quad I_z = 55 \text{ kg m}^2, \\
 &M_s = 40 \text{ Nm}, \quad M_f = 3 \text{ Nm} \\
 &60 \leq x_1 \leq 80, \quad 90 \leq x_2 \leq 110, \quad 1 \leq x_3 \leq 3, \\
 &60 \leq x_4 \leq 1000, \quad 2 \leq x_5 \leq 9, \quad i = 1, 2, 3, 4, 5.
 \end{aligned}$$

Table 17 Comparison of results for multi-plate disc clutch brake problem

Algorithm	ri(×1)	r0(×2)	t(×3)	F(×4)	Z(×5)	Optimal Cost
MBFPA	70	90	1	600	2	0.235242457900804
HHO [11]	69.9999999992493	90	1	1000	3	0.259768993
TLBO [30]	70	90	1	810	3	0.313656
WCA [91]	70	90	1	910	3	0.313656
PVS [92]	70	90	1	980	3	0.31366

Bold values are represent MBPFPA algorithm can get high-precision results

A.3. Pressure vessel design problem

Consider $x = [x_1, x_2, x_3, x_4] = [T_s, T_h, R, L]$.

Minimize

$$f(x) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3.$$

Subject to:

$$\begin{aligned}
 &g_1(x) = -x_1 + 0.0193x_3 \leq 0, \\
 &g_2(x) = -x_3 + 0.00954z_3 \leq 0 \\
 &g_3(x) = -\pi x_3^2 x_4 - \frac{4}{3} \pi x_3^3 + 1,296,000 \leq 0, \\
 &g_4(x) = x_4 - 240 \leq 0 \\
 &\text{Variable range } 0 \leq x_1, x_2 \leq 99; \quad 0 \leq x_3, x_4 \leq 200.
 \end{aligned}$$

A.4. Welded beam design problem

Consider $Z = [z_1, z_2, z_3, z_4] = [h, l, t, b]$.

Minimize $f(Z) = 1.10471z_1^2z_2 + 0.04811z_3z_4(14.0 + z_2)$.

Subject to

$$\begin{aligned}
 &g_1(Z) = \tau(Z) - \tau_{\max} \leq 0, \\
 &g_2(Z) = \sigma(Z) - \sigma_{\max} \leq 0, \\
 &g_3(Z) = \delta(Z) - \delta_{\max} \leq 0 \\
 &g_4(Z) = z_1 - z_4 \leq 0, \\
 &g_5(Z) = P - P_c(Z) \leq 0, \\
 &g_6(Z) = 0.125 - z_1 \leq 0 \\
 &g_7(Z) = 1.10471z_1^2 + 0.04811z_3z_4(14.0 + z_2) - 5.0 \leq 0
 \end{aligned}$$

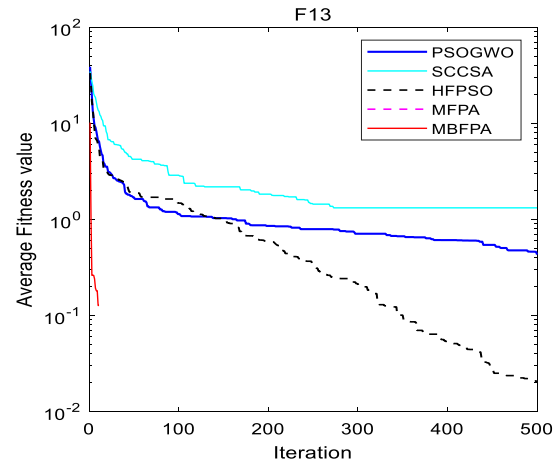


Fig. 20 Convergence graph for Schaffer function

Table 18 Comparison of results for pressure vessel design problem

Algorithms	Ts(×1)	T _h (×2)	R(×3)	L(×4)	Optimal cost
MBFPA	0.77816864137	0.38464916262	40.319618724	200.0	5885.3327736
HHO [11]	0.81758383	0.4072927	42.09174576	176.7196352	6000.46259
GWO [9]	0.812500	0.434500	42.089181	176.758731	6051.5639
GA [49, 93]	0.812500	0.437500	42.097398	176.654050	6059.9463
HPSO [44, 94]	0.812500	0.437500	42.0984	176.6366	6059.7143
GQPSO [50, 95]	0.812500	0.437500	42.0984	176.6372	6059.7208
WEO [51, 96]	0.812500	0.437500	42.098444	176.636622	6059.71
BA [52, 97]	0.812500	0.437500	42.098445	176.636595	6059.7143
MFO [87]	0.8125	0.437500	42.098445	176.636596	6059.7143
CPSO [98]	0.812500	0.437500	42.091266	176.746500	6061.0777
BIANCA [99]	0.812500	0.437500	42.096800	176.65800	6059.9384
MDDE [55, 100]	0.812500	0.437500	42.0968446	176.636047	6059.701660
WOA [10]	0.812500	0.437500	42.0982699	176.638998	6059.7410

Bold values are represent MBPFA algorithm can get high-precision results

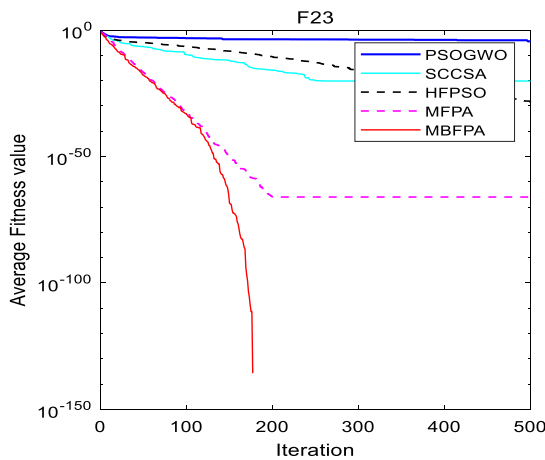


Fig. 21 Convergence graph for Matyas function

Variable range $0.05 \leq z_1 \leq 2.00, 0.25 \leq z_2 \leq 1.30, 2.00 \leq z_3 \leq 15.0$

where $\tau(Z) = \sqrt{\tau'^2 + 2\tau''\tau' \frac{z_2}{2R} + \tau''^2}, \tau' = \frac{P}{\sqrt{2z_1z_2}}, \tau'' = \frac{MR}{J}, M = P(L + \frac{z_2}{2})$

$$R = \sqrt{\frac{z_2^2}{4} + (\frac{z_1+z_3}{2})^2}, J = 2\left\{ \sqrt{2z_1z_2} \left[\frac{z_2^2}{12} + (\frac{z_1+z_3}{2})^2 \right] \right\}$$

$$\sigma(Z) = \frac{6PL}{Ez_4^2}, \delta(Z) = \frac{4PL^3}{Ez_3^3z_4}, P_c(Z) = \frac{4.013E\sqrt{\frac{z_2^2z_4^2}{36}}}{L^2} \left(1 - \frac{z_3}{2L} \sqrt{\frac{E}{4G}} \right)$$

$P = 6000lb, L = 14in, E = 30 \times 10^6psi, G = 12 \times 10^6psi.$

A.5. Speed reducer

Minimize $f(x) = 0.785x_1x_2^2(3.333x_3^2 + 14.9334x_3 - 42.0934) - 1.508x_1(x_6^2 + x_7^2) + 7.4777x_1(x_6^3 + x_7^3) + 1.508x_1(x_4x_6^2 + x_5x_7^2)$

Subject to

$$g_1(X) = \frac{27}{x_1x_2^2x_3} - 1 \leq 0, g_2(X) = \frac{397.5}{x_1x_2x_3^3} - 1 \leq 0$$

$$g_3(X) = \frac{1.93x_4^3}{x_1x_3x_6^4} - 1 \leq 0, g_4(X) = \frac{1.93x_4^3}{x_1x_3x_7^4} - 1 \leq 0$$

$$g_5(X) = \frac{1}{110x_6^3} \sqrt{\left(\frac{745x_4}{x_2x_3}\right)^2 + 16.9 \times 10^6} - 1 \leq 0$$

$$g_6(X) = \frac{1}{85x_7^3} \sqrt{\left(\frac{745x_4}{x_2x_3}\right)^2 + 157.5 \times 10^6} - 1 \leq 0$$

Table 19 Comparison of results for welded beam design problem

Algorithms	h(×1)	l(×2)	t(×3)	b(×4)	Optimal cost
MBFPA	0.205730	3.470473	9.036623	0.205729	1.72485185
HHO [11]	0.204039	3.531061	9.027463	0.206147	1.73199057
SBO [101]	0.21421	3.49240	8.55771	0.22945	1.8496
RANDOM [102]	0.4575	4.7313	5.0853	0.66	4.1185
SIMPLEX [102]	0.2792	5.6256	7.7512	0.2796	2.5307
APPROX [102]	0.2444	6.2189	8.2915	0.2444	2.3815
GSA [21]	0.182129	3.856979	10	0.202376	1.879952
HS [103]	0.2442	6.2231	8.2915	0.2443	2.3807
CDE [104]	0.203137	3.542998	9.033498	0.206179	1.733462

Bold values are represent MBPFA algorithm can get high-precision results

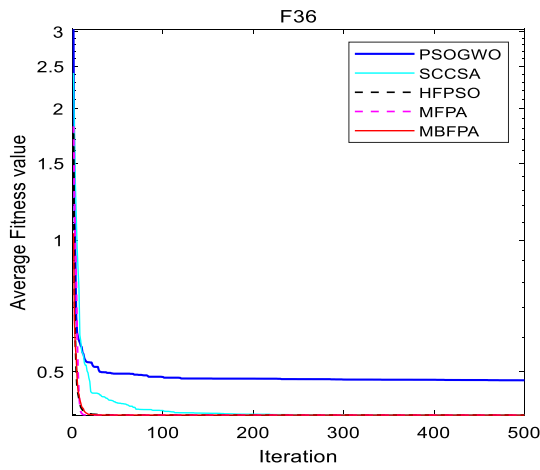


Fig. 22 Convergence graph for Branin function

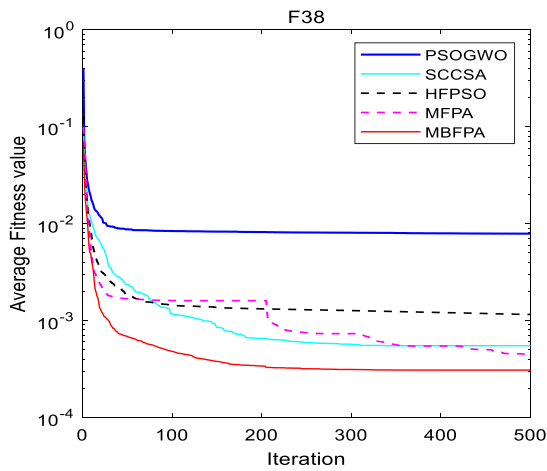


Fig. 23 Convergence graph for Kowalik function

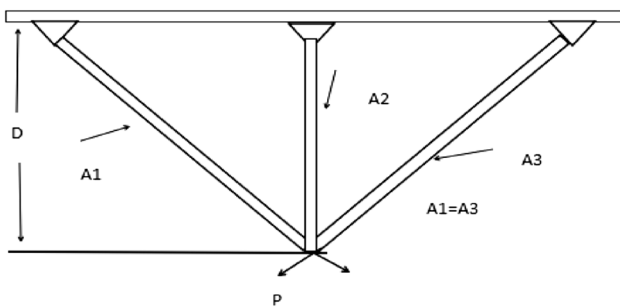


Fig. 24 Three-bar truss design problem

$$g_7(X) = \frac{x_2 x_3}{40} - 1 \leq 0, g_8(X) = \frac{5x_2}{x_1} - 1 \leq 0,$$

$$g_9(X) = \frac{x_1}{12x_2} - 1 \leq 0$$

$$g_{10}(X) = \frac{1.5x_6 + 1.9}{x_4} - 1 \leq 0, g_{11}(X) = \frac{1.1x_7 + 1.9}{x_5} - 1 \leq 0.$$

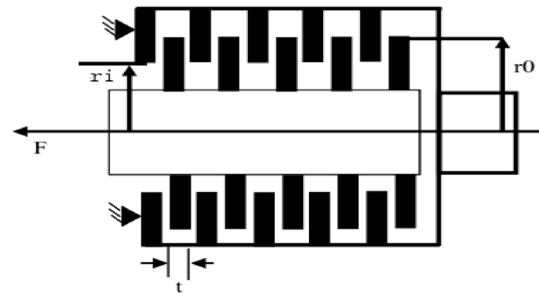


Fig. 25 Multi-plate disc clutch brake problem [30]

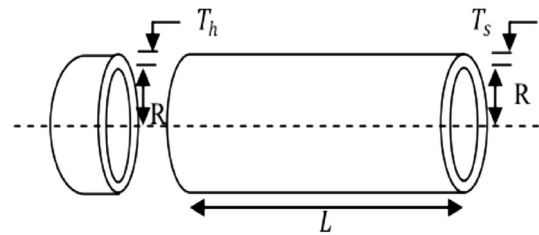


Fig. 26 Pressure vessel design problem [92]

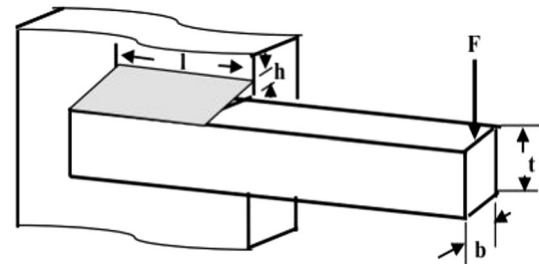


Fig. 27 Welded beam design problem [90]

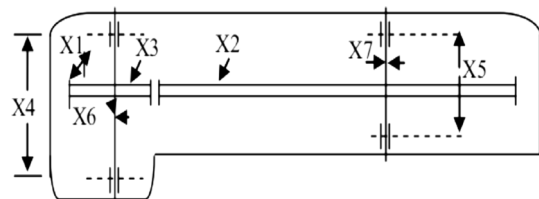


Fig. 28 Speed reducer [90]

where $2.6 \leq x_1 \leq 3.6, 0.7 \leq x_2 \leq 0.8, 17 \leq x_3 \leq 28,$
 $7.3 \leq x_4 \leq 8.3$
 $7.8 \leq x_5 \leq 8.3, 2.9 \leq x_6 \leq 3.9, 5.0 \leq x_7 \leq 5.5.$

Table 20 Comparison of results for speed reducer design problem

Algorithms	×1	×2	×3	×4	×5	×6	×7	Optimal cost
MBFPA	3.500	0.7	17	7.3	7.7153199122	3.35021466	5.28665446	2994.341315
WCA [91]	3.500	0.7	17	7.3	7.715319	3.350214	5.286654	2994.471066
PSODE [88]	3.500	0.7	17	7.3	7.800000	3.350214	5.2866832	2996.348167
MDE [105]	3.50001	0.7	17	7.300156	7.800027	3.350221	5.286685	2996.356689
HEAA [106]	3.500022	0.7	17.000012	7.300427	7.715377	3.350230	5.286663	2994.499107
PVS [92]	3.49999	0.6999	17	7.3	7.8	3.3502	5.2866	2996.3481

Bold values are represent MBPFA algorithm can get high-precision results

References

- Hussain K et al (2019) Metaheuristic research: a comprehensive survey. *Artif Intell Rev* 52:2191–2233
- Blum C et al (2011) Hybrid metaheuristics in combinatorial optimization: a survey. *Appl Soft Comput* 11:4135–4151
- Kennedy J, Eberhart R (1995) Particle swarm optimization. In: *Proceedings of the 1995 IEEE international conference on neural networks*, pp 1942–1948.
- Dorigo M, Birattari M, Stutzle T (2006) Ant colony optimization. *IEEE Comput Intell* 1:28–39
- Basturk B, Karaboga D (2006) An artificial bee colony (ABC) algorithm for numeric function optimization. In: *Proceedings of the IEEE swarm intelligence symposium*, pp 12–14.
- Yang X-S, Deb S (2009) Cuckoo search via Lévy flights. In: *Proceedings of the world congress on nature & biologically inspired computing, NaBIC 2009*, pp 210–14
- Yang XS (2012) Flower pollination algorithm for global optimization. In: *Unconventional computation and natural computation, Lecture notes in computer science, 2012*, 7445, pp 240–249
- Yang X-S (2010) Firefly algorithm, stochastic test functions and design optimisation. *Int J Bio-Inspired Comput* 2:78–84
- Mirjalili S, Mirjalili SM, Lewis A (2014) Grey wolf optimizer. *Adv Eng Softw* 69:46–61
- Mirjalili S, Lewis A (2016) The whale optimization algorithm. *Adv Eng Softw* 95:51–67
- Heidari AA et al (2019) Harris hawks optimization: algorithm and applications. *Future Gener Comput Syst* 97:849–872
- Cheng M-Y, Prayogo D (2014) Symbiotic organisms search: a new metaheuristic optimization algorithm. *Comput Struct* 139:98–112
- Arora S, and S Singh. Butterfly optimization algorithm: a novel approach for global optimization. *Soft Computing* 23.3 (2019): 715–734.
- Goldberg DE, Holland JH (1988) Genetic algorithms and machine learning. *Mach Learn* 3(2):95–99
- Storn R, Price K (1997) Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *J Glob Optim* 11(4):341–359
- Jiao L, Wang L (2000) A novel genetic algorithm based on immunity. *IEEE Trans Syst Man Cybern A Syst Hum* 30:552–561
- De Castro LN, Von Zuben FJ (2000) The clonal selection algorithm with engineering applications. In: *Proceedings of GECCO*, vol 2000.
- Dasgupta D et al (2004) Negative selection algorithm for aircraft fault detection. In: *International conference on artificial immune systems*. Springer, Berlin
- Kelsey J, Timmis J (2003) Immune inspired somatic contiguous hypermutation for function optimisation. *Genetic Evolut Comput Conf*, Springer, Berlin
- Van Laarhoven PJM, Aarts EHL (1987) *Simulated annealing: theory and applications*. Springer, Dordrecht, pp 7–15
- Rashedi E, Nezamabadi-Pour H, Saryzadi S (2009) GSA: a gravitational search algorithm. *Inf Sci* 179:2232–2248
- Shareef H, Ibrahim AA, Mutlag AH (2015) Lightning search algorithm. *Appl Soft Comput* 36:315–333
- Mirjalili S, Mirjalili SM, Hatamlou A (2016) Multi-verse optimizer: a nature-inspired algorithm for global optimization. *Neural Comput Appl* 27(2):495–513
- Abedinpourshotorban H et al (2016) Electromagnetic field optimization: a physics-inspired metaheuristic optimization algorithm. *Swarm Evolut Comput* 26:8–22
- Faramarzi A et al (2019) Equilibrium optimizer: a novel optimization algorithm. *Knowl Based Syst* 105190.
- Lam AYS, Li VOK (2009) Chemical-reaction-inspired metaheuristic for optimization. *IEEE Trans Evolut Comput* 14(3):381–399
- Alatas B (2011) ACROA: artificial chemical reaction optimization algorithm for global optimization. *Expert Syst Appl* 38:13170–13180
- Kumar M, Kulkarni AJ, Satapathy SC (2018) Socio evolution & learning optimization algorithm: a socio-inspired optimization methodology. *Future Gener Comput Syst* 81:252–272
- Kuo HC, Lin CH (2013) Cultural evolution algorithm for global optimizations and its applications. *J Appl Res Technol* 11:510–522
- Rao RV, Savsani VJ, Vakharia DP (2011) Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems. *Comput Aided Design* 43(3):303–315
- Atashpaz-Gargari E, Lucas C (2007) Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition. In: *2007 IEEE congress on evolutionary computation*. IEEE
- Arora S, Singh S (2016) An improved butterfly optimization algorithm for global optimization. *Advanced Science, Engineering and Medicine* 8(9):711–717
- Arora S, Singh S (2017) An improved butterfly optimization algorithm with chaos. *J Intell Fuzzy Syst* 32(1):1079–1088
- Arora S, Singh S (2017) A hybrid optimisation algorithm based on butterfly optimisation algorithm and differential evolution. *Int J Swarm Intell* 3(2–3):152–169
- Arora S, Singh S (2017) An effective hybrid butterfly optimization algorithm with artificial bee colony for numerical optimization. *Int J Interact Multimed Artif Intell* 4(4):14–21
- Arora S, Anand P (2018) Learning automata-based butterfly optimization algorithm for engineering design problems. *Int J Comput Mater Sci Eng* 7(04):1850021
- Arora S, Singh S, Yetilmesoy K (2018) A modified butterfly optimization algorithm for mechanical design optimization problems. *J Braz Soc Mech Sci Eng* 40(1):21

38. Singh B, Anand P (2018) A novel adaptive butterfly optimization algorithm. *Int J Comput Mater Sci Eng* 7(04):1850026
39. Li G, Shuang F, Zhao P et al (2019) An improved butterfly optimization algorithm for engineering design problems using the cross-entropy method. *Symmetry* 11(8):1049
40. Sharma S, Saha AK (2020) m-MBOA: a novel butterfly optimization algorithm enhanced with mutualism scheme. *Soft Comput* 24:4809–4827
41. Sharma S, Saha AK, Ramasamy V et al (2020) hBOSOS: an ensemble of butterfly optimization algorithm and symbiosis organisms search for global optimization. In: *Advanced computing and intelligent engineering*. Springer, Singapore, vol. 1089, pp 579–588
42. Arora S, Anand P (2019) Binary butterfly optimization approaches for feature selection. *Expert Syst Appl* 116:147–160
43. Jalali SMJ, Ahmadian S, Kebria PM, et al (2019) Evolving artificial neural networks using butterfly optimization algorithm for data classification. In: *International conference on neural information processing*. Springer, Cham, pp 596–607.
44. Du P et al (2019) Container throughput forecasting using a novel hybrid learning method with error correction strategy. *Knowl-Based Syst* 182:104853
45. Kisi O et al (2019) Drought forecasting using novel heuristic methods in a semi-arid environment. *J Hydrol* 578:124053
46. Priyadharshini P, Prakashraj K, Padmapriya S (2019) Improved butterfly optimization algorithm using local search operator for capacitated vehicle routing problem. *Inter Res J Eng Technol (IRJET)* 6(4):983–988
47. Abdul-Rashid R, Alawode BO (2019) Robustness evaluation of the butterfly optimization algorithm on a control system. *arXiv preprint arXiv:1912.00185*
48. El Hajjami L, Mellouli EM, Berrada M (2019) Optimal PID control of an autonomous vehicle using butterfly optimization algorithm BOA. In: *Proceedings of the 4th international conference on big data and internet of things*, pp 1–5.
49. Arora S, Singh S (2017) Node localization in wireless sensor networks using butterfly optimization algorithm. *Arab J Sci Eng* 42(8):3325–3335
50. Wang R, Zhou Y (2014) Flower pollination algorithm with dimension by dimension improvement. *Math Probl Eng* 481791, 9 pages
51. Zhao C, Zhou Y (2016) A complex encoding flower pollination algorithm for global numerical optimization. In: *International conference on intelligent computing*. Springer, Cham, pp. 667–678.
52. Zhou Y, Wang R, Luo Q (2016) Elite opposition-based flower pollination algorithm. *Neurocomputing* 188:294–310
53. Nabil E (2016) A modified flower pollination algorithm for global optimization. *Expert Syst Appl* 57:192–203
54. Singh D, Singh U, Salgotra R (2018) An extended version of flower pollination algorithm. *Arab J Sci Eng* 43(12):7573–7603
55. Lei M, Zhou Y, Luo Q (2019) Enhanced metaheuristic optimization: wind-driven flower pollination algorithm. *IEEE Access* 7:111439–111465
56. Pauline O, Meng OK, Kiong SC (2017) An improved flower pollination algorithm with chaos theory for function optimization. *AIP Conf Proc AIP Publ LLC* 1870(1):050012
57. Pathak P, Mahajan K (2015) A pollination based optimization for load balancing task scheduling in cloud computing. *Int J Adv Res Comput Sci* 6(7).
58. Jensi R, Jiji GW (2015) Hybrid data clustering approach using k-means and flower pollination algorithm. *arXiv preprint arXiv:1505.03236*.
59. Agarwal P, Mehta S (2016) Enhanced flower pollination algorithm on data clustering. *Int J Comput Appl* 38(2–3):144–155
60. Wang R, Zhou Y, Qiao S et al (2016) Flower pollination algorithm with bee pollinator for cluster analysis. *Inf Process Lett* 116(1):1–14
61. Harikrishnan R, Jawahar Senthil Kumar V, Sridevi Ponmalar P (2015) Nature inspired flower pollen algorithm for WSN localization problem. *ARPN J Eng. Appl Sci* 10(5):2122–2125
62. Goyal S, Patterh MS (2015) Flower pollination algorithm based localization of wireless sensor network. In: *2015 2nd international conference on recent advances in engineering & computational sciences (RAECS)*. IEEE, pp 1–5.
63. Binh HTT, Hanh NT, Dey N (2018) Improved cuckoo search and chaotic flower pollination optimization algorithm for maximizing area coverage in wireless sensor networks. *Neural Comput Appl* 30(7):2305–2317
64. Bensouyad M, Saidouni DE (2015) A hybrid discrete flower pollination algorithm for graph coloring problem. In: *Proceedings of the international conference on engineering & MIS 2015*, pp 1–6.
65. Chakraborty D, Saha S, Maity S (2015) Training feedforward neural networks using hybrid flower pollination-gravitational search algorithm. In: *2015 International conference on futuristic trends on computational analysis and knowledge management (ABLAZE)*. IEEE, pp 261–266.
66. Chiroma H, Khan A, Abubakar AI et al (2016) A new approach for forecasting OPEC petroleum consumption based on neural network train by using flower pollination algorithm. *Appl Soft Comput* 48:50–58
67. Velamuri S, Sreejith S, Ponnambalam P (2016) Static economic dispatch incorporating wind farm using flower pollination algorithm. *Perspect Sci* 8:260–262
68. Gonidakis D (2016) Application of flower pollination algorithm to multi-objective environmental/economic dispatch. *Int J Manag Sci Eng Manag* 11(4):213–221
69. Putra PH, Saputra TA (2016) Modified flower pollination algorithm for nonsmooth and multiple fuel options economic dispatch. In: *2016 8th international conference on information technology and electrical engineering (ICITEE)*. IEEE, pp 1–5.
70. Shilaja C, Ravi K (2017) Optimization of emission/economic dispatch using euclidean affine flower pollination algorithm (eFPA) and binary FPA (BFPA) in solar photo voltaic generation. *Renew Energy* 107:550–566
71. Abdel-Baset M, Ibrahim MH (2015) An improved flower pollination algorithm for ratios optimization problems. *Appl Math Inf Sci Lett Int J* 3(2):83–91
72. Gautam U, Malmathanraj R, Srivastav C (2015) Simulation for path planning of autonomous underwater vehicle using flower pollination algorithm, genetic algorithm and Q-learning. In: *2015 international conference on cognitive computing and information processing (CCIP)*. IEEE, pp 1–5
73. Zhou Y, Wang R (2016) An improved flower pollination algorithm for optimal unmanned undersea vehicle path planning problem. *Int J Pattern Recognit Artif Intell* 30(04):1659010
74. Wang R, Zhou Y, Zhao C et al (2015) A hybrid flower pollination algorithm based modified randomized location for multi-threshold medical image segmentation. *Bio Med Mater Eng* 26(s1):S1345–S1351
75. Doğan B, Ölmez T (2015) A new metaheuristic for numerical function optimization: vortex search algorithm. *Inf Sci* 293:125–145
76. Dokeroglu T, Sevinc E, Kucukyilmaz T et al (2019) A survey on new generation metaheuristic algorithms. *Comput Ind Eng* 137:106040
77. Ezugwu AE, Prayogo D (2019) Symbiotic organisms search algorithm: theory, recent advances and applications. *Expert Syst Appl* 119:184–209

78. Rashedi E, Rashedi E, Nezamabadi-pour H (2018) A comprehensive survey on gravitational search algorithm. *Swarm Evolut Comput* 41:141–158
79. Rubinstein RY (1997) Optimization of computer simulation models with rare events. *Eur J Oper Res* 99(1):89–112
80. Ram JP, Babu TS, Dragicevic T et al (2017) A new hybrid bee pollinator flower pollination algorithm for solar PV parameter estimation. *Energy Convers Manag* 135:463–476
81. Zimmerman DW, Zumbo BD (1993) Relative power of the Wilcoxon test, the Friedman test, and repeated-measures ANOVA on ranks. *J Exp Educ* 62(1):75–86
82. Singh N, Singh SB (2017) Hybrid algorithm of particle swarm optimization and grey wolf optimizer for improving convergence performance. *J Appl Math* 2017:1–14
83. Khalilpourazari S, Pasandideh SHR (2019) Sine-cosine crow search algorithm: theory and applications. *Neural Comput Appl* 1–18. <https://doi.org/10.1007/s00521-019-04530-0>. in press
84. Luo J, Shi B (2019) A hybrid whale optimization algorithm based on modified differential evolution for global optimization problems. *Appl Intell* 49(5):1982–2000
85. Zhang M, Luo W, Wang X (2008) Differential evolution with dynamic stochastic selection for constrained optimization. *Inform Sci* 178:3043–3074
86. Saremi S, Mirjalili S, Lewis A (2017) "Grasshopper optimisation algorithm: theory and application. *Adv Eng Softw* 105:30–47
87. Mirjalili S (2015) Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm. *Knowl Based Syst* 89:228–249
88. Liu H, Cai Z, Wang Y (2010) Hybridizing particle swarm optimization with differential evolution for constrained numerical and engineering optimization. *Appl Soft Comput* 10:629–640
89. Mirjalili S et al (2017) Salp swarm algorithm: a bio-inspired optimizer for engineering design problems. *Adv Eng Softw* 114:163–191
90. Sadollah A et al (2013) Mine blast algorithm: a new population based algorithm for solving constrained engineering optimization problems. *Appl Soft Comput* 13(5):2592–2612
91. Zheng YJ (2015) Water wave optimization: a new nature-inspired metaheuristic. *Comput Oper Res* 55:1–11
92. Savsani P, Savsani V (2016) Passing vehicle search (PVS): A novel metaheuristic algorithm. *Appl Math Model* 40:3951–3978
93. Deb K (1991) Optimal design of a welded beam via genetic algorithms. *AIAA J* 29(11):2013–2015
94. He Q, Wang L (2007) A hybrid particle swarm optimization with a feasibility-based rule for constrained optimization. *Appl Math Comput* 186:1407–1422
95. Yao W (2014) Genetic quantum particle swarm optimization algorithm for solving traveling salesman problems. In: *Fuzzy information & engineering and operations research & management*. Springer, Berlin, pp 67–74
96. Kaveh A, Bakhshpoori T (2016) Water evaporation optimization: a novel physically inspired optimization algorithm. *Comput Struct* 167:69–85
97. Gandomi AH et al (2013) Bat algorithm for constrained optimization tasks. *Neural Comput Appl* 22:1239–1255
98. He Q, Wang L (2007) An effective co-evolutionary particle swarm optimization for constrained engineering design problems. *Eng Appl Artif Intell* 20(1):89–99
99. Montemurro M, Vincenti A, Vannucci P (2013) The automatic dynamic penalisation method (ADP) for handling constraints with genetic algorithms. *Comput Methods Appl Mech Eng* 256:70–87
100. Mezura-Montes E et al (2007) Multiple trial vectors in differential evolution for engineering design. *Eng Opt* 39(5):567–589
101. Moosavi SHS, Bardsiri VK (2017) Satin bowerbird optimizer: a new optimization algorithm to optimize ANFIS for software development effort estimation. *Eng Appl Artif Intell* 60:1–15
102. Ragsdell KM, Phillips DT (1976) Optimal design of a class of welded structures using geometric programming, pp 1021–1025.
103. Lee KS, Geem ZW (2004) A new structural optimization method based on the harmony search algorithm. *Comput Struct* 82:781–798
104. Huang F, Wang L, He Q (2007) An effective co-evolutionary differential evolution for constrained optimization. *Appl Math Comput* 186(1):340–356
105. Mezura-Montes E, Velázquez-Reyes J, Coello CAC (2006) Modified differential evolution for constrained optimization. In: 2006 IEEE international conference on evolutionary computation. IEEE, pp 25–32
106. Wang Y et al (2009) Constrained optimization based on hybrid evolutionary algorithm and adaptive constraint-handling technique. *Struct Multidiscip Optim* 37:395–413
107. Braekers K, Ramaekers K, Van Nieuwenhuysse I (2016) The vehicle routing problem: state of the art classification and review. *Comput Ind Eng* 99:300–313
108. Caceres-Cruz J, Arias P, Guimarans D et al (2014) Rich vehicle routing problem: Survey. *ACM Comput Surv (CSUR)* 47(2):1–28
109. McCulloch WS, Pitts W (1943) A logical calculus of the ideas immanent in nervous activity. *Bull Math Biophys* 5(4):115–133
110. Aljarah I, Faris H, Mirjalili S (2018) Optimizing connection weights in neural networks using the whale optimization algorithm. *Soft Comput* 22(1):1–15
111. Zhang Y, Jin Z, Chen Y (2020) Hybrid teaching-learning-based optimization and neural network algorithm for engineering design optimization problems. *Knowl Based Syst* 187:104836
112. Zhang JR, Zhang J, Lok TM et al (2007) A hybrid particle swarm optimization-back-propagation algorithm for feedforward neural network training. *Appl Math Comput* 185(2):1026–1037
113. Alba E, Chicano JF (2004) Training neural networks with GA hybrid algorithms. In: *Genetic and evolutionary computation conference*. Springer, Berlin, pp 852–863.
114. Sexton RS, Dorsey RE, Johnson JD (1999) Optimization of neural networks: a comparative analysis of the genetic algorithm and simulated annealing. *Eur J Oper Res* 114(3):589–601
115. Sexton RS, Gupta JND (2000) Comparative evaluation of genetic algorithm and backpropagation for training neural networks. *Inf Sci* 129(1–4):45–59
116. Siddique MNH, Tokhi MO (2001) Training neural networks: backpropagation vs. genetic algorithms. In: *IJCNN'01. International joint conference on neural networks*. Proceedings (Cat. No. 01CH37222), vol 4. IEEE, pp 2673–2678.
117. Whitley D, Starkweather T, Bogart C (1990) Genetic algorithms and neural networks: Optimizing connections and connectivity. *Parallel Comput* 14(3):347–361
118. Hansen N, Müller SD, Koumoutsakos P (2003) Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (cma-es). *Evolut Comput* 11(1):1–18
119. Ilonen J, Kamarainen JK, Lampinen J (2003) Differential evolution training algorithm for feed-forward neural networks. *Neural Process Lett* 17(1):93–105
120. Slowik A, Bialko M (2008) Training of artificial neural networks using differential evolution algorithm. In: 2008 conference on human system interactions. IEEE, pp 60–65
121. Wdaa ASI, Sttar A (2008) Differential evolution for neural networks learning enhancement. *Universiti Teknologi Malaysia, Johor Bahru*
122. Wienholt W (1993) Minimizing the system error in feedforward neural networks with evolution strategy. In: *International*

- conference on artificial neural networks. Springer, London, pp 490–493.
123. Heidari AA, Faris H, Mirjalili S et al (2020) Ant lion optimizer: theory, literature review, and application in multi-layer perceptron neural networks, *Nature-Inspired Optimizers*. Cham, Springer, pp 23–46
 124. Derrac J, García S, Molina D et al (2011) A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm Evolut Comput* 1(1):3–18
 125. Faris H, Aljarah I, Mirjalili S (2016) Training feedforward neural networks using multi-verse optimizer for binary classification problems. *Appl Intell* 45(2):322–332
 126. Wolpert DH, Macready WG (1997) No free lunch theorems for optimization. *IEEE Trans Evol Comput* 1(1):67–82
 127. Tian J, Sun C, Tan Y et al (2020) Granularity-based surrogate-assisted particle swarm optimization for high-dimensional expensive optimization. *Knowl Based Syst* 187:104815
 128. Zheng Y, Fu X, Xuan Y (2019) Data-driven optimization based on random forest surrogate. In: 2019 6th international conference on systems and informatics (ICSAI). IEEE, pp 487–491

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.