Complex & Intelligent Systems (2022) 8:65–82
https://doi.org/10.1007/s40747-021-00346-5

ORIGINAL ARTICLE

An enhanced chimp optimization algorithm for continuous optimization domains

Heming Jia[1,2] · Kangjian Sun[2] · Wanying Zhang[2] · Xin Leng[2]

Received: 26 November 2020 / Accepted: 17 March 2021 / Published online: 7 April 2021

Abstract

Chimp optimization algorithm (ChOA) is a recently proposed metaheuristic. Interestingly, it simulates the social status relationship and hunting behavior of chimps. Due to the more flexible and complex application fields, researchers have higher requirements for native algorithms. In this paper, an enhanced chimp optimization algorithm (EChOA) is proposed to improve the accuracy of solutions. First, the highly disruptive polynomial mutation is used to initialize the population, which provides the foundation for global search. Next, Spearman's rank correlation coefficient of the chimps with the lowest social status is calculated with respect to the leader chimp. To reduce the probability of falling into the local optimum, the beetle antennae operator is used to improve the less fit chimps while gaining visual capability. Three strategies enhance the exploration and exploitation of the native algorithm. To verify the function optimization performance, EChOA is comprehensively analyzed on 12 classical benchmark functions and 15 CEC2017 benchmark functions. Besides, the practicability of EChOA is also highlighted by three engineering design problems and training multilayer perceptron. Compared with ChOA and five state-of-the-art algorithms, the statistical results show that EChOA has strong competitive capabilities and promising prospects.

Keywords Chimp optimization algorithm · Highly disruptive polynomial mutation · Spearman's rank correlation coefficient · Beetle antennae operator · Continuous optimization domains

Introduction

During the fourth industrial revolution, people have entered the artificial intelligence (AI) age. Faced with complex research branches and massive data, it is essential to design effective tools to explore valuable information. As one of the elements in the development of AI, algorithms regularly promote the optimization and innovation of technique. Brute-force search [1] finds the best solution possible but is very low even in two or three dimensions. Hill-climbing [2] usually finds the local optimum instead of the global warm. These algorithms are hard to solve exactly classical NP-complete problems such as Hamiltonian cycle problem [3], traveling salesman problem [4], and coloring problem [5], etc. This is also the reason that metaheuristic algorithms (MAs) enter the blowout period. For a specific issue, heuristic algorithms provide estimated solutions in feasible time and space. To adapt to a wide range of the issues, MAs are proposed, which are independent of the existence of the problem. If necessary, some fine-tuning of internal parameters will adapt to the current issue [6].

When designing a MA, two components should be considered: exploration and exploitation. The convergence speed and solution accuracy of the algorithm mainly depend on the balance level between the two components. In the initial phase, a well-organized optimizer should highly explore the search space to find diverse solutions. After a fine transition, the algorithm often uses local information to generate better solutions, which are usually in the neighborhood of the current solutions [7, 8]. More than hundreds of MAs have so far developed. Based on different design inspirations, MAs can be divided into four categories. The physics-inspired MAs use physical laws, which mainly include gravitational search algorithm (GSA) [9], multi-verse optimizer (MVO) [10], and thermal exchange optimization (TEO) [11]. The evolution-inspired MAs use Darwinian natural selection. The most well-known algorithms are genetic algorithm

✉ Heming Jia
jiaheminglucky99@126.com

[1] College of Information Engineering, Sanming University, Sanming 365004, Fujian, China

[2] College of Mechanical and Electrical Engineering, Northeast Forestry University, Harbin 150040, Heilongjiang, China

(GA) [12] and differential evolution (DE) [13]. The swarm-inspired MAs mainly simulates the collective behavior of creatures in nature. The most classical algorithm belongs to particle swarm optimization (PSO) [14]. swarm intelligence (SI) optimization algorithms proposed in recent years also include ant lion optimizer (ALO) [15], whale optimization algorithm (WOA) [16], and marine predators algorithm (MPA) [17], etc. The last category is the human behavior-inspired MAs represented by teaching–learning-based optimization (TLBO) [18].

MAs have their importance in various fields of computational sciences. However, most MAs suffer from premature convergence, stuck in local optimum, poor convergence speed, etc. No free lunch (NFL) theorem [19] turns out that the world is dominated by real-world problems without a known provably efficient algorithm. Therefore, many researchers have been motivated by the NFL theorem to propose novel algorithms. Some of them have demonstrated very well performance for optimization problems. However, some algorithms rely only on attractive metaphors and are criticized for lack of novelty [20, 21]. In 2020, Chimp optimization algorithm was proposed by Khishe and Mosavi. An overview of ChOA can be found in Sect. “Chimp optimization algorithm”. One of the main motivations in this paper is to propose an enhanced ChOA version (named EChOA), which is created to improve the performance of the algorithm. It is not a novel SI algorithm but an enhancement of the existing one inspired by experimental studies on its behavior. The other main motivation is that the AI-enabled application is further developed. To the best of our knowledge, it is the first time that the enhanced ChOA is used to train multilayer perceptron. The main contributions of this paper are stated more clearly as follows:

1. HDPM helps further to explore the regions and boundaries of the initial space.
2. Spearman’s rank correlation coefficient refines candidate solutions that need to be improved.
3. Beetle antennae operator helps to avoid the local optimum and provide better exploitation.
4. The performance of EChOA is evaluated on 27 benchmark functions, three engineering design problems, and training multilayer perceptron.
5. EChOA outperforms contemporary optimization algorithms.

Section "Related works" outlines some related works. Section "Preliminaries" introduces preliminaries, including ChOA and three strategies. In Sect. "Proposed algorithm", the proposed algorithm is described in detail. In Sect. "Function optimization experiments", the experimental results of benchmark functions are given. In Sect. "Engineering optimization experiments" and Sect. "Training multilayer perceptron experiments", the research aspects of solving engineering design problems and training multilayer perceptron are discussed and analyzed, respectively. Finally, Sect. "Conclusions and future research" summarizes the work.

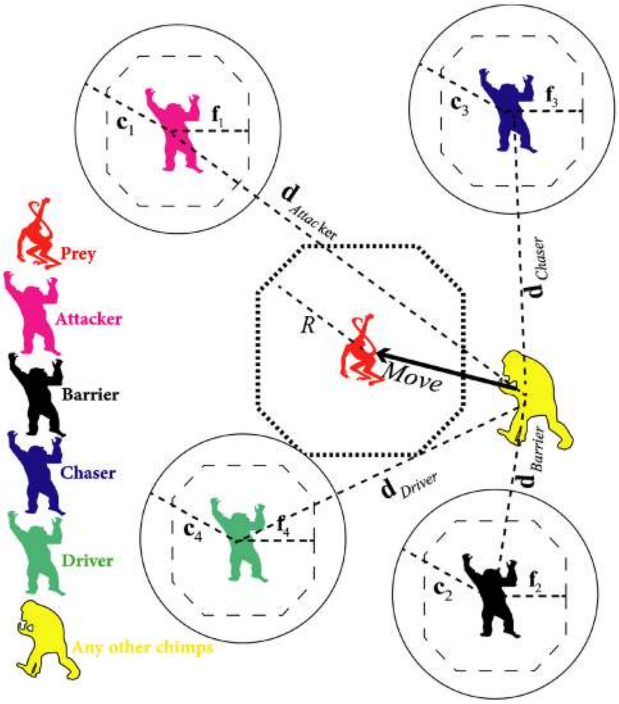

Fig. 1 Process of position update [37]

Related works

In the introduction, some well-known algorithms with different categories are listed. In this section, the recent works done about ChOA and other MAs are briefed hereafter. Some of these algorithms in the literature are then used as the comparative algorithms in this paper.

Khishe and Mosavi used ChOA to train a neural network to classify the underwater acoustical dataset. The proposed method obtained better classification accuracy with respect to the ion motion algorithm, gray wolf optimization, and PSO-GSA [22]. It is one of the few works about the new application of ChOA. The MA-ALS algorithm combined GA and adaptive local search was proposed by Arab et al., which resulted in improved accuracy and convergence speed. The performance was evaluated on function optimization and optimal controller design [23]. Li et al. also developed a novel hybrid algorithm based on PSO and artificial bee colony (ABC) and tested it on 13 high-dimensional benchmark functions. The efficiency and robustness of PS-ABC were verified in detail [24]. As another variant of PSO, Berkan Aydilek also attempted to hybridize firefly optimization algorithm (FOA) into PSO in 2018 [25]. Through analysis, we believe that the main motivation of [23–25] is to combine one algorithm for the global exploration phase and the other for locally refining exploitation by applying reasonable laws.

In 2017, Mirjalili et al. embedded 10 chaotic maps and an adaptive transformation mechanism into GSA to improve the performance. Experimental results demonstrated that the sinusoidal map is the most appropriate for this specific problem [26]. Meanwhile, Khishe et al. also used four chaotic maps to improve the performance of the stochastic fractal search (SFS) algorithm. The proposed algorithm trained multi-layer perceptron and applied it to the sonar dataset. The capability to address high dimensional problems was one of its advantages [27]. To accomplish the same task, an improved biogeography-based optimization (BBO) algorithm based on mutation operators was proposed in [28] and implemented on hardware. Jia et al. proposed a multi-strategy emperor penguin optimizer (MSEPO) based on HDPM, Levy flight, and thermal exchange operator. The effectiveness of the suggested three strategies was analyzed and verified in the experiments. It was an effective multilevel thresholding segmentation algorithm for color satellite images [29].

Khishe and Mosavi redesigned seven spiral shapes in the WOA algorithm. Different spiral shapes can directly affect the convergence behavior of the algorithm. This metaheuristic trainer based on the proposed algorithm was applied to data classification [30]. MAs applied in this field also include salp swarm algorithm (SSA) [31]. Mousavi et al. recently improved WOA by splitting sub-populations and embedding fractional chaotic maps and applied this enhanced algorithm (EWOA) to identify parameters of wind–diesel power systems [32]. Shokri-Ghaleh et al. proposed an improved version of the cuckoo optimization algorithm (ULCOA) for non-linear field calibration problems in 2020 [33]. In 2020, Ma et al. improved the step size of the beetle antennae search algorithm to optimize the Huber Loss function [34]. The artificial bee colony (ABC) algorithm fused three kinds of knowledge to improve search capability. This idea can be regarded as an optimization framework for other MAs. [35]. In the improved grey wolf optimizer (I-GWO) algorithm, the dimension learning-based hunting (DLH) search strategy was used to enhance the information sharing among wolfs [36].

Based on the above in-depth literature review, the recent trend for improvement is to combine the two algorithms or apply some strategies to the entire solution. However, this paper attempts to apply the search gain to only certain components of the solution besides introducing the mutation operator. Generally, it can enhance the result more finely.

Preliminaries

Chimp optimization algorithm

ChOA is a novel MA based on swarm intelligence proposed by Khishe and Mosavi in 2020 [37]. Its intuitive background originates from the hunting behavior of chimps. Chimps perform different actions according to the division of labor to find the prey. The standard ChOA algorithm divides the chimp group into four types: attacker, barrier, chaser, and driver. Among them, the attacker is the leader of the population. The other three types of chimps assisted in hunting, and their status decreased in turn. The mathematical model is briefly described as following. Equations (1) and (2) is used to update the position of the chimp. Figure 1 visualizes this effect:

$$\begin{aligned} X_1(t+1) &= X_{\text{Attacker}}(t) - a_1 \cdot d_{\text{Attacker}} \\ X_2(t+1) &= X_{\text{Barrier}}(t) - a_2 \cdot d_{\text{Barrier}} \\ X_3(t+1) &= X_{\text{Chaser}}(t) - a_3 \cdot d_{\text{Chaser}} \\ X_4(t+1) &= X_{\text{Driver}}(t) - a_4 \cdot d_{\text{Driver}} \end{aligned} \tag{1}$$

$$X_{\text{chimp}}(t+1) = \frac{X_1 + X_2 + X_3 + X_4}{4} \tag{2}$$

where t represents the number of the current iteration, the position of the chimp is updated according to the four types of position stored (X_{Attacker}, X_{Barrier}, X_{Chaser}, and X_{Driver}). The dynamic coefficient a and vector d are expressed in Eq. (3):

Fig. 2 Beetle with long antennae [43]

$$\begin{aligned} a_1 &= 2 \cdot f_1 \cdot r_1 - f_1, \quad d_{\text{Attacker}} = \left| c \cdot X_{\text{Attacker}}(t) - m \cdot X(t) \right| \\ a_2 &= 2 \cdot f_2 \cdot r_1 - f_2, \quad d_{\text{Barrier}} = \left| c \cdot X_{\text{Barrier}}(t) - m \cdot X(t) \right| \\ a_3 &= 2 \cdot f_3 \cdot r_1 - f_3, \quad d_{\text{Chaser}} = \left| c \cdot X_{\text{Chaser}}(t) - m \cdot X(t) \right| \\ a_4 &= 2 \cdot f_4 \cdot r_1 - f_4, \quad d_{\text{Driver}} = \left| c \cdot X_{\text{Driver}}(t) - m \cdot X(t) \right| \end{aligned} \tag{3}$$

where the coefficient f decreases nonlinearly from 2.5 to 0 with the lapse of iteration. $c = 2r_2$. r_1 and r_2 are random numbers in [0, 1]. m is a chaotic map vector. Assuming the probability μ is a random number in [0, 1], the chaotic model is used for position updating when $\mu \geq 0.5$, as shown in Eq. (4). Otherwise, Eq. (2) is still executed. Algorithm 1 shows the pseudo-code of the ChOA algorithm:

$$X_{\text{chimp}}(t + 1) = \text{Chaotic\_value} \tag{4}$$

Algorithm 1 Pseudo-code of ChOA

```
Initialize the population size and the maximum number of iterations
Initializes positions of chimps
Calculate the fitness of each chimp
Select attacker, barrier, chaser, and driver.
While $t$<the maximum number of iterations
    For each chimp
        If $\mu$<0.5
            Update the position of the current chimp using Eq. (2)
        Else if
            Update the position of the current chimp using Eq. (4)
        End if
    End for
    Update $f$, $c$, $m$, $a$ and $d$
    Calculate the fitness of all chimp
    Update $X_{Attacker}$, $X_{Barrier}$, $X_{Chaser}$, and $X_{Driver}$
    $t$=$t$+1
End while
Return $X_{Attacker}$
```

Highly disruptive polynomial mutation

Common mutation operators include random mutation, non-uniform mutation, and polynomial mutation. For the traditional polynomial mutation (PM), the mutation has no effect

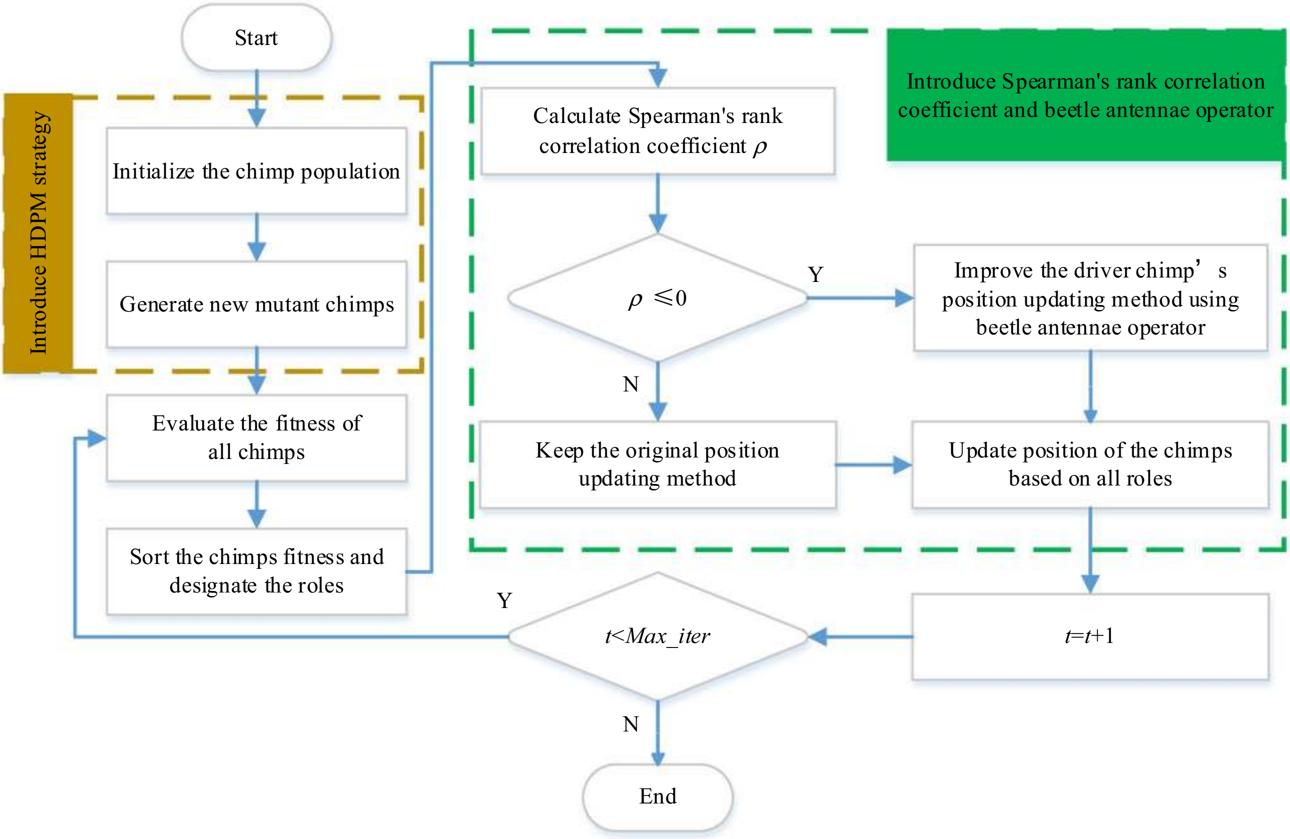
Start
Introduce HDPM strategy
Initialize the chimp population
Generate new mutant chimps
Evaluate the fitness of all chimps
Sort the chimps fitness and designate the roles
Calculate Spearman's rank correlation coefficient ρ
Introduce Spearman's rank correlation coefficient and beetle antennae operator
ρ ≤0
Y
N
Improve the driver chimp' s position updating method using beetle antennae operator
Keep the original position updating method
Update position of the chimps based on all roles
t<Max_iter
Y
N
t=t+1
End

Fig. 3 Flowchart of EChOA

when the variable is on the boundary. HDPM improves this disadvantage [38, 39]. The form of the operator is in Eq. (5):

$$X_{\text{new}} = X + \delta_k \cdot (ub - lb) \tag{5}$$

where *ub* and *lb* represent the upper and lower boundaries of the search space, respectively. *X* is the parent, X_{new} is the offspring. The coefficient δ_k is calculated by Eqs. (6)–(8):

$$\delta_1 = \frac{X - lb}{ub - lb} \tag{6}$$

$$\delta_2 = \frac{ub - X}{ub - lb} \tag{7}$$

$$\delta_k = \begin{cases} [2r + (1-2r)\cdot(1-\delta_1)^{\eta_m+1}]^{\frac{1}{\eta_m+1}} - 1, & \text{if } r \le 0.5 \\ 1 - [2(1-r) + 2(r-0.5)\cdot(1-\delta_2)^{\eta_m+1}]^{\frac{1}{\eta_m+1}}, & \text{otherwise} \end{cases} \tag{8}$$

where *r* is a random number in [0, 1], η_m is the mutation index. It can be observed from Eq. (8) that it can still make full use of the whole search space even if the variable is on one of the boundaries. This advantage maintains the diversity of candidate solutions.

Spearman's rank correlation coefficient

Spearman's rank correlation coefficient is a non-parametric index used to measure the statistical correlation between the two series [40]. The two series u_i and v_i are sorted. The ranks u_i' and v_i' represent the positions of sorted u_i and v_i, respectively. The difference between them is $d_i = u_i' - v_i', i = 1, 2, ...n$. Spearman;s rank correlation coefficient ρ is calculated as follows. Its value is in the range of [− 1, 1]. 0 means that the two series are no correlation. (0, 1] means positive correlation, [− 1, 0) means negative correlation, and the higher value means a stronger correlation:

$$\rho = 1 - \frac{6\sum d_i^2}{n\cdot(n^2-1)} \tag{9}$$

where *n* represents the dimension of the series.

Beetle antennae operator

When a beetle (see Fig. 2) is preying, it receives the food smell of near-area using two antennas and finds the area with the strongest smell. If the antennae on one side receives a stronger concentration of smell, the beetle will turn to the same side; otherwise, it will turn to the other side. Considering the sense of smell, the beetle antennae search algorithm is formally proposed by Jiang and Li [41]. First, the random direction vector $\vec{b}$ is normalized to the following form:

$$\vec{b} = \frac{\text{rnd}(n,1)}{\|\text{rnd}(n,1)\|} \tag{10}$$

where rnd(.) represents a random function, and *n* represents the dimension of search space. The two search behaviors in Eq. (11) simulates the beetle exploring the left and right areas using two antennas:

$$\begin{aligned} X_r(t) &= X(t) + d(t)\cdot\vec{b} \\ X_l(t) &= X(t) - d(t)\cdot\vec{b} \end{aligned} \tag{11}$$

where *X* represents the original position of the beetle, X_r represents the position obtained by the beetle exploring the right areas, X_l represents the position obtained by the beetle searching the left areas, *d* represents the distance between two antennas. The update rule is as follows:

$$d(t) = \frac{\delta(t)}{C} \tag{12}$$

where δ represents the step size, *C* represents the attenuation rate, $C = 2$ [42]. δ is calculated in Eq. (13):

Table 1 Parameter settings of different algorithms

Algorithm	Parameters	Value
EO	Parameter λ	[0, 1]
	Generation probability *GP*	0.5
	Parameter a_1	2
	Parameter a_2	1
LFD	Threshold	2
	Scalar *CSV*	0.5
	Index β	1.5
	Random number a_1	10
	Random number a_2	0.00005
	Random number a_3	0.005
	Parameter δ_1	0.9
	Parameter δ_2	0.1
HFPSO	Random number *a*	0.2
	Attractiveness B_0	2
	Distance *r*	1
	Acceleration coefficient c_1	1.49445
	Acceleration coefficient c_2	1.49445
	Inertia weight $w_{\max}$	0.9
	Inertia weight $w_{\min}$	0.5
CGSA10	Initial value of chaotic map x_0	0.7
	Gravitational constant G_0	100
	Constant *a*	20
I-GWO	Parameter *a*	[0, 2]

Table 2 Details of 12 classical benchmark functions

Function	Class	Range
$F_{01}(x) = \sum_{i=1}^{n} x_i^2$	Unimodal	[− 100, 100]
$F_{02}(x) = \sum_{i=1}^{n} \lvert x_i\rvert + \prod_{i=1}^{n} \lvert x_i\rvert$	Unimodal	[− 10, 10]
$F_{03}(x) = \sum_{i=1}^{n} \left(\sum_{j-1}^{i} x_j\right)^2$	Unimodal	[− 100, 100]
$F_{04}(x) = max\{\lvert x_i\rvert, 1 < i < n\}$	Unimodal	[− 100, 100]
$F_{05}(x) = \sum_{i=1}^{n-1} \left[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2\right]$	Unimodal	[− 30, 30]
$F_{06}(x) = \sum_{i=1}^{n} \left(\left[x_i + 0.5\right]\right)^2$	Unimodal	[− 100, 100]
$F_{07}(x) = \sum_{i=1}^{n} ix_i^4 + \text{random}[0,1)$	Unimodal	[− 1.28, 1.28]
$F_{09}(x) = \sum_{i=1}^{n} \left[x_i^2 - 10\cos\left(2\pi x_i\right) + 10\right]$	Multimodal	[− 5.12, 5.12]
$F_{10}(x) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos\left(2\pi x_i\right)\right) + 20 + e$	Multimodal	[− 32, 32]
$F_{11}(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	Multimodal	[− 600, 600]
$F_{12}(x) = \frac{\pi}{n}\left\{10\sin\left(\pi y_1\right) + \sum_{i=1}^{n-1}\left(y_i - 1\right)^2[1 + 10sin^2\left(\pi y_{i+1}\right) + \left(y_n - 1\right)^2\right\} + \sum_{i=1}^{n} u\left(x_i, 10, 100, 4\right)$	Multimodal	[− 50, 50]
$y_i = 1 + \frac{x_i + 1}{4} u\left(x_i, a, k, m\right) = \begin{cases} k\left(x_i - a\right)^m & x_i > a \\ 0 & -a < x_i < a \\ k\left(-x_i - a\right)^m & x_i < a \end{cases}$		
$F_{13}(x) = 0.1\{sin^2\left(3\pi x_1\right) + \sum_{i=1}^{n}\left(x_i - 1\right)^2\left[1 + sin^2\left(3\pi x_i + 1\right)\right] + \left(x_n - 1\right)^2$ $\times\left[1 + sin^2\left(2\pi x_n\right)\right]\} + \sum_{i=1}^{n} u\left(x_i, 5, 100, 4\right)$	Multimodal	[− 50, 50]

Table 3 Settings of different strategy combinations

	HDPM	Correlation coefficient and beetle antennae operator
ChOA	0	0
ChOA1	1	0
ChOA2	0	1
EChOA	1	1

$$\delta(t) = K \cdot \delta(t - 1) \tag{13}$$

where K is the attenuation rate of δ, K=0.95 [41]. Based on the above search behaviors, the new position update model of the beetle is shown in Eq. (14):

$$X(t + 1) = X(t) + \delta(t) \cdot \vec{b} \cdot \text{sign}\left(f(X_r(t)) - f(X_l(t))\right) \tag{14}$$

where sign(.) represents a sign function.

Table 4 Average fitness results of different combinations on 12 classical benchmark functions

F	ChOA	ChOA1	ChOA2	EChOA
F01	7.7223E − 06	9.3568E − 17	1.6848E − 22	**1.2202E − 36**
F02	3.2295E − 05	6.9785E − 12	1.4127E − 17	**6.5794E − 23**
F03	1.0266E+02	6.6010E − 09	1.2706E − 12	**4.3166E − 19**
F04	2.8411E − 01	1.3391E − 10	3.8679E − 13	**2.5706E − 16**
F05	2.8850E+01	1.6184E+01	1.4163E+01	**1.0437E+01**
F06	3.6396E+00	1.9711E+00	1.2517E+00	**1.1092E+00**
F07	1.4896E − 03	9.3673E − 04	5.8464E − 05	**1.0987E − 06**
F09	1.4008E+01	5.7691E+00	1.7842E − 01	**1.4185E − 01**
F10	1.9962E+01	2.2204E − 10	**1.5099E − 14**	1.9481E − 14
F11	2.6092E − 02	1.8164E − 02	4.2539E − 03	**3.1093E − 03**
F12	4.7654E − 01	3.7679E − 05	5.2757E − 10	**7.4552E − 12**
F13	2.7403E+00	1.2777E − 02	9.0927E − 07	**1.0024E − 10**

Proposed algorithm

Although the native ChOA algorithm divides chimps into different social levels for cooperative hunting, it lacks population diversity in the initialization phase. Besides, it also falls in the local optimum easily in the last detailed search phase. In this section, three strategies, including HDPM,

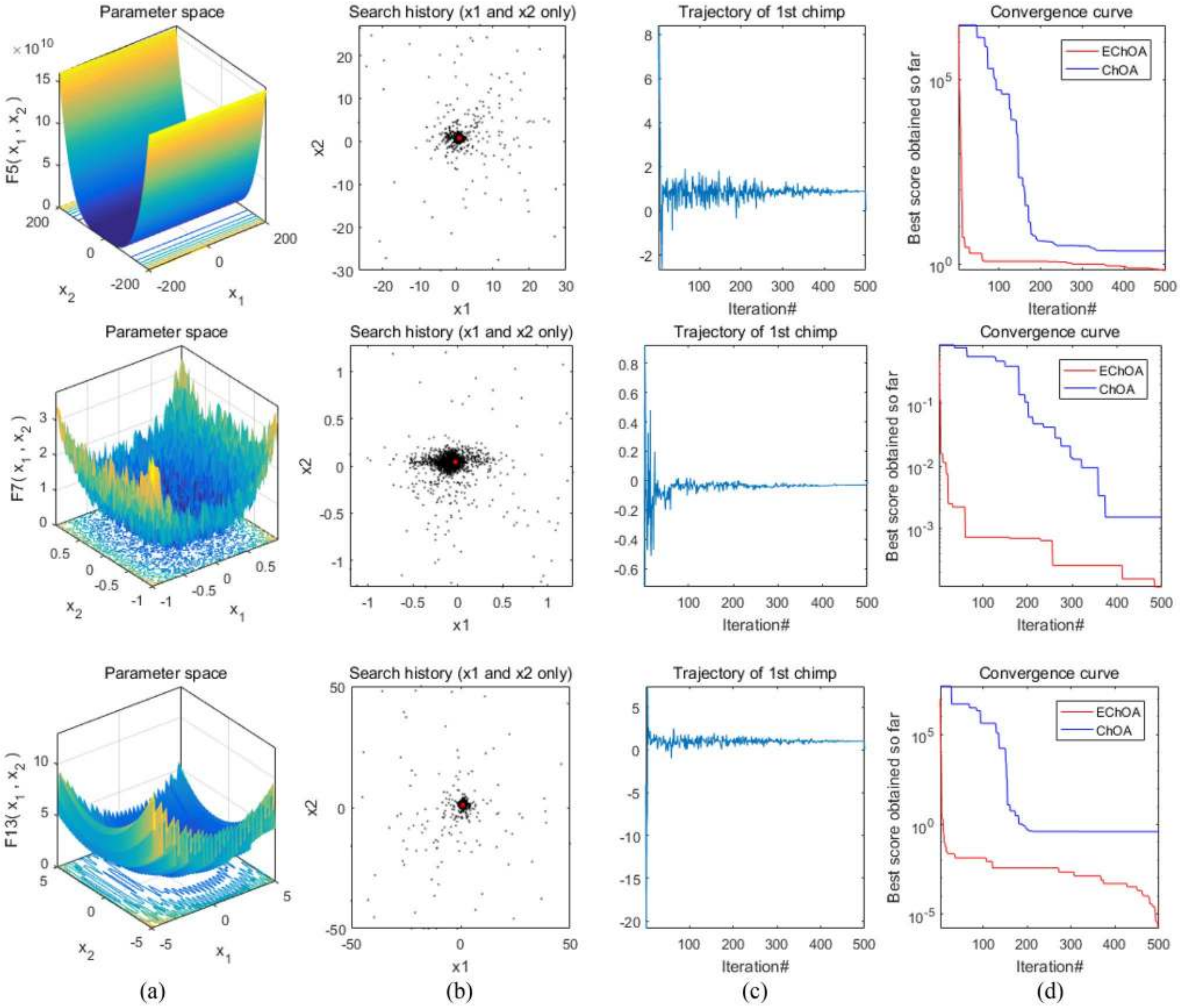

Fig. 4 **a** Parameter space. **b** Search history of EChOA. **c** Trajectory of EChOA in the first chimp. **d** Convergence curve of ChOA and EChOA

Spearman's rank correlation coefficient, and beetle antennae operator are introduced to enhance the performance of the native ChOA algorithm. The enhanced version is named EChOA. The pseudo-code is given in Algorithm 2, and the flowchart is shown in Fig. 3. The details of the proposed algorithm are explained as follows.

First, the HDPM strategy is introduced to enhance the population diversity in the initialization phase. A well-organized optimizer should achieve an appropriate balance between exploration and exploitation. Further, high exploration steps are carried out at the beginning of the search, and more exploitation steps are often required in the last phase. By introducing the HDPM strategy, the exploration capability of the algorithm is enhanced. Second, the Spearman's rank correlation coefficient of the driver chimps with respect to the attacker chimp is calculated. In the native ChOA algorithm, the attacker chimp as the leader plays an important role in guiding the population. However, it can be observed from Eq. (2) that the final position is updated not only by the attacker chimp but also by the driver chimps with less social level and fit. Therefore, by calculating the Spearman's rank correlation coefficient, it can determine whether the driver chimp is near or far away from the attacker chimp. For the driver chimp far away from the attacker chimp, we change the position updating method to improve the fit. Finally, two series are negatively correlated or uncorrelated when the Spearman's rank correlation coefficient is less than or equal to 0. Meanwhile, the less fit chimp (the driver chimp) is improved. The beetle antennae operator is introduced. As described in Sect. "Beetle antennae operator", the less fit chimp gains visual capability. The position updating method of chimp is improved using Eq. (14). It utilizes its

Table 5 Average fitness results of ChOA and EChOA in different dimensions on 12 classical benchmark functions

F	30		50		100	
	ChOA	EChOA	ChOA	EChOA	ChOA	EChOA
F01	7.7223E − 06	**1.2202E − 36**	5.2398E − 03	**2.2325E − 35**	9.5014E − 01	**8.3822E − 33**
F02	3.2295E − 05	**6.5794E − 23**	5.5773E − 03	**1.4383E − 21**	6.4973E − 02	**5.0171E − 20**
F03	1.0266E + 02	**4.3166E − 19**	2.1025E + 02	**6.7592E − 19**	9.0444E + 02	**1.1324E − 18**
F04	2.8411E − 01	**2.5706E − 16**	2.8994E + 00	**3.3147E − 16**	8.2501E + 01	**4.3428E − 15**
F05	2.8850E + 01	**1.0437E + 01**	4.8972E + 01	**4.8627E + 01**	1.7901E + 02	**9.7893E + 01**
F06	3.6396E + 00	**1.1092E + 00**	8.6031E + 00	**4.0003E + 00**	2.3056E + 01	**1.2507E + 01**
F07	1.4896E − 03	**1.0987E − 06**	1.3708E − 03	**1.5895E − 06**	4.0216E − 02	**4.6171E − 05**
F09	1.4008E + 01	**1.4185E − 01**	3.2399E + 01	**5.6843E − 01**	4.0426E + 01	**1.1369E + 00**
F10	1.9962E + 01	**1.9481E − 14**	1.9965E + 01	**6.4837E − 12**	1.9971E + 01	**1.4317E − 11**
F11	2.6092E − 02	**3.1093E − 03**	1.0259E − 01	**6.4109E − 03**	1.0591E − 01	**9.9920E − 03**
F12	4.7654E − 01	**7.4552E − 12**	8.3770E − 01	**1.8173E − 11**	8.7288E − 01	**4.8866E − 11**
F13	2.7403E + 00	**1.0024E − 10**	4.7009E + 00	**2.8465E − 10**	9.9282E + 00	**7.5741E − 08**

Table 6 Fitness results of different algorithms on 12 classical benchmark functions

F		EO	LFD	HFPSO	CGSA10	I-GWO	ChOA	EChOA
F01	Avg	1.6170E − 23	3.0993E − 07	3.2812E − 05	9.4112E + 04	1.7304E − 28	7.7223E − 06	**1.2202E − 36**
	Std	7.8302E − 23	1.0756E − 07	2.5400E − 05	1.4041E + 04	2.4004E − 28	1.2221E − 05	**6.3326E − 36**
F02	Avg	1.9691E − 12	3.2558E − 04	5.2264E − 03	4.1778E − 02	7.5942E − 18	3.2295E − 05	**6.5794E − 23**
	Std	9.3125E − 12	4.8672E − 05	2.6546E − 03	1.4479E − 01	6.4458E − 18	3.4737E − 05	**6.5418E − 23**
F03	Avg	1.8194E − 18	1.5230E − 06	1.4441E + 02	1.6653E + 06	2.9177E − 03	1.0266E + 02	**4.3166E − 19**
	Std	6.8144E − 18	4.0094E − 07	7.4368E + 01	1.8665E + 06	1.0877E − 02	2.4079E + 02	**1.5968E − 18**
F04	Avg	8.1523E − 14	3.4102E − 04	2.0783E + 00	9.5673E + 01	2.8119E − 05	2.8411E − 01	**2.5706E − 16**
	Std	3.1401E − 13	4.9296E − 05	6.1350E − 01	2.5393E + 00	3.7238E − 05	2.5059E − 01	**8.9657E − 16**
F05	Avg	1.2440E + 01	2.8051E + 01	5.1327E + 01	2.7193E + 08	2.4392E + 01	2.8850E + 01	**1.0437E + 01**
	Std	1.4225E + 01	**1.4211E − 01**	4.0559E + 01	6.8410E + 07	7.9887E − 01	2.8108E − 01	8.2122E − 01
F06	Avg	**7.1125E − 10**	1.8184E + 00	6.1290E − 01	9.2573E + 04	4.0925E − 02	3.6396E + 00	1.1092E + 00
	Std	**1.0317E − 09**	3.9499E − 01	1.0977E + 00	1.5578E + 04	9.1425E − 02	3.2955E − 01	4.3338E − 01
F07	Avg	9.7758E − 05	1.0628E + 00	1.4883E − 02	1.0395E − 01	2.5625E − 03	1.4896E − 03	**1.0987E − 06**
	Std	1.1316E − 04	5.2436E − 01	5.9072E − 03	4.0837E − 02	1.0006E − 03	1.4926E − 03	**7.9915E − 07**
F09	Avg	2.9849E + 00	**7.4709E − 06**	4.8306E + 01	1.8241E + 01	2.0584E + 01	1.4008E + 01	1.4185E − 01
	Std	8.9546E + 00	**1.8776E − 05**	1.5342E + 01	5.0232E + 00	7.2316E + 00	1.5233E + 01	6.0519E − 01
F10	Avg	7.8859E − 12	1.3036E − 04	5.5938E − 01	2.0703E + 01	6.0811E − 14	1.9962E + 01	**1.9481E − 14**
	Std	4.0699E − 11	1.9759E − 05	6.8395E − 01	2.1803E − 01	1.1036E − 14	1.1443E − 03	**3.6327E − 15**
F11	Avg	**0.0000E + 00**	8.5356E − 07	1.1200E − 02	9.0754E + 02	3.2914E − 03	2.6092E − 02	3.1093E − 03
	Std	**0.0000E + 00**	2.2361E − 07	1.5621E − 02	1.4253E + 02	6.3726E − 03	4.6616E − 02	7.3349E − 03
F12	Avg	1.0639E − 10	7.7341E − 01	2.0065E − 02	9.3354E + 08	1.4692E − 05	4.7654E − 01	**7.4552E − 12**
	Std	1.3557E − 10	1.2477E − 01	5.5352E − 02	3.0820E + 08	4.3331E − 05	1.9997E − 01	**2.9305E − 12**
F13	Avg	7.0539E − 10	2.9328E + 00	1.8018E − 02	1.8117E + 09	1.0566E − 01	2.7403E + 00	**1.0024E − 10**
	Std	2.1688E − 09	1.3236E − 01	3.5997E − 02	5.2650E + 08	1.2856E − 01	1.4713E − 01	**2.8722E − 11**

left and right eyes to observe the environment on both sides and further determines the direction of the next step. The purpose of this improvement is to prevent the driver chimp from falling in the local optimum due to poor performance. It is considered to boost the exploitation trend.

Algorithm 2 Pseudo-code of EChOA

```
Initialize the population size $N$ and the maximum number of iterations $Max\_iter$
Initializes positions of chimps $X$
Generate mutant positions of chimps using Eq. (5)                                   //Introduce HDPM//
Calculate the fitness of each chimp
Select attacker, barrier, chaser, and driver
While $t<Max\_iter$
  For $i$=driver to $N$
    For $j$=1 to $n$
      Calculate Spearman's rank correlation coefficient $\rho$ using Eq. (9)   //Introduce Spearman's rank correlation coefficient//
      If $\rho\leq0$
        Update the position of $X_i$ using Eq. (14)                                 //Introduce beetle antennae operator//
      End if
    End for
  End for
  For $i$=1 to $N$
    If $\mu<0.5$
      Update the position of the current chimp using Eq. (2)
    Else if
      Update the position of the current chimp using Eq. (3)
    End if
  End for
Update $f$, $c$, $m$, $a$ and $d$
Calculate the fitness of all chimp
Update $X_{Attacker}, X_{Barrier}, X_{Chaser}, and\ X_{Driver}$
$t=t+1$
End while
Return $X_{Attacker}$
```

Function optimization experiments

In this section, the performance of the proposed EChOA algorithm is evaluated based on the function optimization. The native ChOA algorithm and five state-of-the-art algorithms from 2017 to 2021, including EO [44], LFD [45], HFPSO [25], CGSA10 [26], and I-GWO [36], are used as competitors to verify the improvements of the proposed algorithm. Some of these comparison algorithms are the native ones recently proposed, and some are improved ones with good performance. Table 1 shows the parameter settings of different algorithms. The values used are the same as those in the corresponding references. Besides, the maximum of iterations is 500, and the population size is 30. All the experimental series were carried out on MATLAB R2016a, and the computer was configured as Intel(R) Core (TM) i5-1035G1 CPU @1.00 GHz 1.19 GHz, using Microsoft Windows 10 system. Each experiment is run 30 times independently.

Table 7 Details of 15 CEC2017 benchmark functions

No.	Name	Class	$F_{\min}$
F04	SRF Rosenbrock’s function	Multimodal	400
F05	SRF Rastrigin’s function	Multimodal	500
F06	SRF expanded Scaffer’s F6 function	Multimodal	600
F07	SRF Lunacek Bi-Rastrigin function	Multimodal	700
F08	SRF noncontinuous Rastrigin’s function	Multimodal	800
F09	SRF Levy function	Multimodal	900
F11	HF1($N=3$)	Hybrid	1100
F16	HF6($N=4$)	Hybrid	1600
F17	HF6($N=5$)	Hybrid	1700
F19	HF6($N=5$)	Hybrid	1900
F21	CF1($N=3$)	Composition	2100
F25	CF5($N=5$)	Composition	2500
F26	CF6($N=5$)	Composition	2600
F27	CF7($N=6$)	Composition	2700
F29	CF9($N=3$)	Composition	2900

*N represents the basic number of functions

Impact of introduced strategies

Due to the mutual influence among the strategies, in this subsection, 12 classical benchmark functions [46] are used to verify the effectiveness of the introduced strategies. The formula of functions, the class, and the range of variables are given in Table 2. The number of variables and the global optimum is all 30 and 0, respectively. The settings of the samples of different strategy combinations are shown in Table 3. Here, 1 and 0 are used to represent whether the strategy is introduced, respectively. The average fitness results obtained are shown in Table 4. Note that the best values has been highlighted in boldface in all following tables. Based on these results, the three enhanced versions all outperform the native algorithm. EChOA outperforms ChOA1 for 12 classical benchmark functions, and ChOA2 for 11 functions. On the one hand, ChOA1 and ChOA2, respectively, reach a higher level in terms of search breadth and search accuracy compared with ChOA. On the other hand, the performance of EChOA is comprehensively improved with the effective combination of the introduced strategies. After validation, EChOA is selected as the final optimized version of the native algorithm.

Table 8 Fitness results of different algorithms on 15 CEC2017 benchmark functions

F		EO	LFD	HFPSO	CGSA10	I-GWO	ChOA	EChOA
F04	Avg	3.3818E+03	8.5581E+02	4.4101E+02	2.3377E+04	4.0384E+02	6.5063E+02	**4.0338E+02**
	Std	1.3435E+03	1.1135E+02	4.5452E+01	1.5951E+04	1.2163E+00	1.8229E+02	**1.1019E+00**
F05	Avg	6.5871E+02	5.8589E+02	5.5292E+02	8.2374E+02	5.1198E+02	5.6072E+02	**5.0707E+02**
	Std	3.2326E+01	1.3714E+01	1.8402E+01	1.2098E+02	6.3950E+00	1.3426E+01	**5.7011E+00**
F06	Avg	6.9142E+02	6.4641E+02	6.2194E+02	7.6954E+02	6.0014E+02	6.4028E+02	**6.0012E+02**
	Std	1.0649E+01	7.2723E+00	1.3521E+01	4.0641E+01	**3.6350E−02**	1.0790E+01	3.9394E+00
F07	Avg	8.9003E+02	8.4993E+02	7.6343E+02	1.6580E+03	**7.2761E+02**	8.0675E+02	7.4266E+02
	Std	**4.3783E+00**	1.4870E+01	1.7263E+01	2.5632E+02	9.8944E+00	1.7866E+01	1.3513E+01
F08	Avg	9.0325E+02	8.8154E+02	8.3952E+02	**8.0632E+02**	8.0787E+02	8.4792E+02	8.1919E+02
	Std	1.6861E+01	9.9282E+00	1.4438E+01	8.2518E+01	6.5295E+00	9.7080E+00	**5.9720E+00**
F09	Avg	2.7393E+03	1.8209E+03	1.1775E+03	1.3836E+04	**9.0001E+02**	1.3950E+03	9.3767E+02
	Std	3.2260E+02	2.1994E+02	3.0667E+02	6.7106E+03	**1.0486E−02**	3.0565E+02	5.9709E+01
F11	Avg	3.4421E+04	1.7873E+03	1.1959E+03	4.4827E+08	1.1051E+03	1.3990E+03	**1.1044E+03**
	Std	2.2575E+04	2.8833E+02	5.2380E+01	6.6524E+08	2.1797E+00	1.0773E+02	**1.8980E+00**
F16	Avg	2.6478E+03	2.0525E+03	1.9629E+03	2.7739E+04	1.6066E+03	1.9608E+03	**1.6050E+03**
	Std	2.1697E+02	1.1639E+02	1.5877E+02	4.2371E+04	4.2364E+00	1.4878E+02	**3.2319E+00**
F17	Avg	2.2014E+03	1.8927E+03	1.8248E+03	2.6971E+05	1.7394E+03	1.7883E+03	**1.7357E+03**
	Std	8.9954E+01	4.9418E+01	8.3997E+01	3.8333E+05	1.5305E+01	1.6272E+01	**1.4215E+01**
F19	Avg	9.9979E+07	2.6170E+04	3.5177E+04	2.1767E+10	**1.9581E+03**	2.4709E+04	2.5950E+04
	Std	1.3030E+08	3.6799E+04	3.8300E+04	2.4308E+10	**3.8432E+01**	8.6563E+03	7.0906E+04
F21	Avg	2.4567E+03	2.3088E+03	2.3377E+03	3.0230E+03	2.2819E+03	2.3213E+03	**2.2200E+03**
	Std	2.6438E+01	3.6122E+01	4.7750E+01	7.0541E+02	4.9986E+01	5.5371E+01	**2.5314E+01**
F25	Avg	4.5200E+03	3.2740E+03	2.9581E+03	1.0893E+04	2.9053E+03	3.0282E+03	**2.8382E+03**
	Std	3.9231E+02	1.1144E+02	5.0193E+01	4.5375E+03	1.6562E+01	7.2071E+01	**1.4367E+01**
F26	Avg	4.9095E+03	3.6120E+03	3.2147E+03	8.5021E+03	2.8935E+03	3.9488E+03	**2.8389E+03**
	Std	4.0007E+02	1.6963E+02	3.4227E+02	3.0947E+03	2.5392E+01	3.5213E+02	**1.9843E+01**
F27	Avg	3.3716E+03	3.1539E+03	3.1538E+03	7.1836E+03	3.0896E+03	3.1115E+03	**3.0089E+03**
	Std	2.1780E+02	2.3373E+01	3.9370E+01	3.1198E+03	**4.9590E−01**	1.8961E+01	2.9929E+01
F29	Avg	3.9498E+03	3.4109E+03	3.3864E+03	9.7546E+06	3.1677E+03	3.3832E+03	**3.1063E+03**
	Std	1.0809E+02	6.4239E+01	9.3976E+01	3.2010E+07	2.0089E+01	9.3948E+01	**1.7743E+01**

Qualitative analysis

Figure 4 shows the qualitative results of testing on F05, F07, and F13. Figure 4b shows the two-dimensional scatter plot of the search history of EChOA. Through observation, it can be seen that chimps are scattered in the entire search space in the early phase, and concentrated rapidly in the later phase. When addressing different functions, the density of distribution is significantly different. This demonstrates that EChOA can change the balance between exploration and exploitation. Figure 4c shows the trajectory of the first chimp. There are some sudden changes in the early optimization phase. It reveals that EChOA does not fall into the early local optimum and has better exploration capability. The amplitude of these fluctuations gradually decreases with the lapse of iteration. Finally, the chimp tends to be stable and converges to the optimal position. Figure 4d shows the convergence curve of ChOA and EChOA. For functions such as F05, F07, and F13, EChOA have a better fitness curve that lasts 500 iterations than ChOA. EChOA can find a smaller solution, which demonstrates that the excellent convergence capability is one of the features for EChOA.

Analysis of scalability

Scalability is an important indicator of a new-designed algorithm. By testing the different dimensions of the benchmark function, we can effectively judge the influence of dimension expansion on the execution efficiency of the algorithm. According to experience, 12 classical benchmark functions are tested on three dimensions of 30, 50, and 100. The experimental results are made in Table 5. It can be seen from the data comparison that as the number of dimension increases, the optimization performance of two algorithms gradually decreases. The main reason is that the dimension of data reflects the number of factors to be optimized. The larger

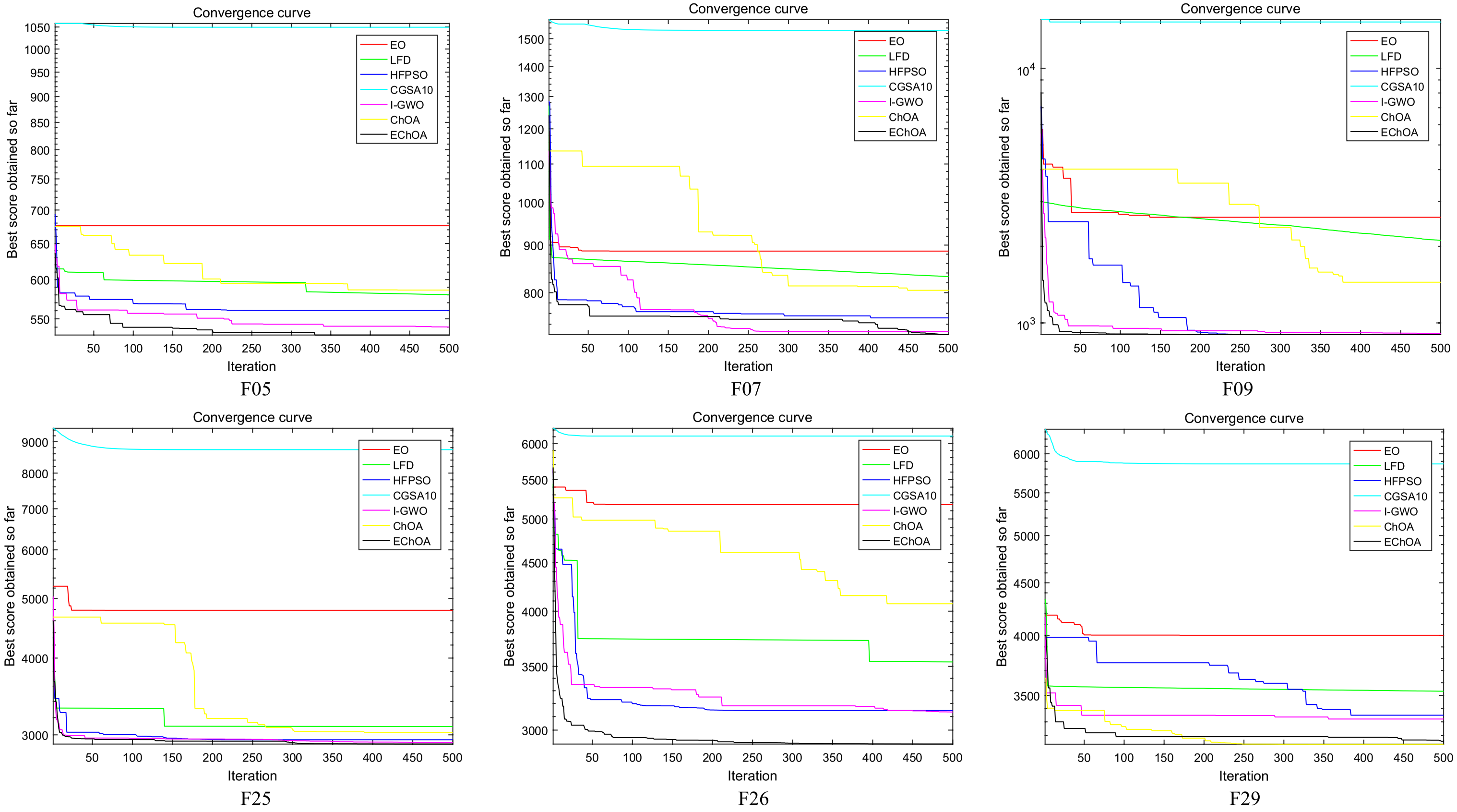

Fig. 5 Convergence curves of different algorithms on F05, F07, F09, F25, F26, and F29

Table 9 Runtime results of different algorithms on 15 CEC2017 benchmark functions

F	EO	LFD	HFPSO	CGSA10	I-GWO	ChOA	EChOA
F04	1.2579E − 01	2.2316E+00	1.3538E − 01	9.4234E − 01	5.2318E − 01	1.0168E+00	1.9123E+00
F05	1.8745E − 01	2.2287E+00	9.3462E − 02	6.2194E − 01	5.9093E − 01	7.0633E − 01	1.9154E+00
F06	3.6564E − 01	2.3182E+00	1.1842E − 01	5.8494E − 01	6.7541E − 01	8.6802E − 01	1.7614E+00
F07	3.0977E − 01	2.5804E+00	1.2970E − 01	6.6852E − 01	5.5228E − 01	9.7160E − 01	1.0894E+00
F08	3.2849E − 01	2.2765E+00	1.3049E − 01	7.5798E − 01	5.4430E − 01	1.0369E+00	1.1677E+00
F09	1.5083E − 01	2.4313E+00	1.2641E − 01	6.3035E − 01	5.4478E − 01	9.5658E − 01	1.3126E+00
F11	3.3954E − 01	2.5419E+00	1.1912E − 01	7.0190E − 01	5.5882E − 01	8.8230E − 01	1.0391E+00
F16	1.3180E − 01	2.2316E+00	1.4275E − 01	7.8232E − 01	5.4188E − 01	1.0679E+00	8.3449E − 01
F17	1.5389E − 01	2.2803E+00	9.3727E − 02	8.5148E − 01	5.7463E − 01	7.0120E − 01	8.0784E − 01
F19	3.6973E − 01	2.7213E+00	1.5701E − 01	9.7306E − 01	7.3966E − 01	1.0056E+00	8.8346E − 01
F21	2.1897E − 01	2.6679E+00	1.4510E − 01	8.6626E − 01	5.8018E − 01	1.1361E+00	9.7240E − 01
F25	3.9999E − 01	2.3264E+00	9.6235E − 02	5.1479E − 01	5.9742E − 01	7.1238E − 01	8.4313E − 01
F26	4.5253E − 01	2.3452E+00	1.1605E − 01	6.2516E − 01	6.2773E − 01	7.5772E − 01	9.6169E − 01
F27	4.3989E − 01	2.3677E+00	1.2916E − 01	8.7839E − 01	6.2384E − 01	9.3619E − 01	1.4167E+00
F29	3.9911E − 01	2.5651E+00	1.4910E − 01	8.1732E − 01	6.1274E − 01	1.0805E+00	8.3510E − 01

Table 10 Experimental results of Wilcoxon rank-sum test

Algorithm	Multimodal	Hybrid	Composition
EChOA vs. EO	1.3203E − 03	1.1429E − 03	9.5238E − 04
EChOA vs. LFD	1.4289E − 03	1.1429E − 03	9.5238E − 04
EChOA vs. HFPSO	6.9913E − 03	4.5132E − 03	2.2222E − 03
EChOA vs. CGSA10	4.1126E − 04	2.8571E − 04	1.5873E − 04
EChOA vs. I-GWO	1.0000E − 02	8.8571E − 03	5.4762E − 03
EChOA vs. ChOA	4.8485E − 03	6.8571E − 03	1.5079E − 03

Table 11 Experimental results of Friedman test

Algorithm	Multimodal	Hybrid	Composition
EO	5.5000	6.0000	5.6000
LFD	4.4167	4.2500	3.9000
HFPSO	4.0000	4.1250	4.0000
CGSA10	6.5000	7.0000	7.0000
I-GWO	1.6667	1.7500	2.2000
ChOA	4.2500	3.1250	4.0000
EChOA	1.5500	1.6000	1.3000
Chi-sq	50.4643	41.0893	47.3571
p-value	3.7939E − 09	2.7807E − 07	1.5880E − 08

Table 12 Settings of different control parameters

	η_{m}	K
EChOA1	0.2	0.70
EChOA2	0.4	0.80
EChOA3	0.6	0.90
EChOA4	0.8	0.93
EChOA5	1	0.95
EChOA6	2	0.97

the dimension, the more complex the search space. On the other hand, Table 5 also show that the optimization results of EChOA are always better than that of ChOA, and the gap of optimization effect becomes more and more significant as the number of dimension increases. When addressing high-dimensional problems, EChOA can maintain well exploitation while exploring certain optimality.

Comparison with other algorithms on classical benchmark functions

Table 6 gives the average (Avg) and the standard deviation (Std) of fitness obtained by each algorithm on 12 classical benchmark functions. It can be seen from the table above that EChOA gives the best results in general. For example, in the case of F01, the average fitness values are 1.6170E − 23, 3.0993E − 07, 3.2812E − 05, 9.4112E+04, 1.7304E − 28, 7.7223E − 06, and 1.2202E − 36 for EO, LFD, HFPSO, CGSA10, I-GWO, ChOA, and EChOA, respectively. To verify the stability of the proposed algorithm, the Std indicator is also used. A lower value of Std indicates better stability. From Table 6, it can be found that EChOA gives lower values as compared to other algorithms, which shows the better consistency and stability of the proposed algorithm. In essence, the results of unimodal benchmark functions reflect the high exploitation of the proposed algorithm. Considering the characteristics of multimodal benchmark functions, it can be said that EChOA has a robust capability to avoid the local optimum.

Table 13 Average ranking of Friedman test

	EChOA1	EChOA2	EChOA3	EChOA4	EChOA5	EChOA6
Rank	2.9211	2.8520	2.3800	2.7734	**2.0860**	2.1566

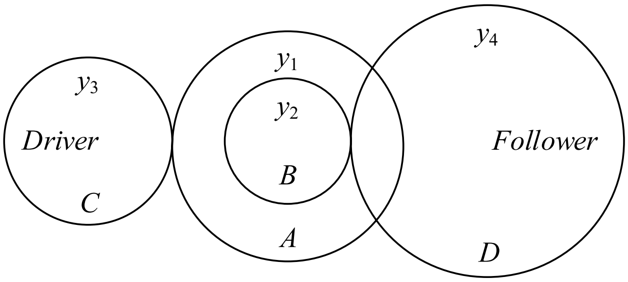

Fig. 6 Gear train [49]

Table 14 Experimental results of gear train design

Algorithm	y_1	y_2	y_3	y_4	F_{min}
EO	48.6528	17.5152	22.3078	54.1230	1.1661E − 10
LFD	31.9406	12.5234	22.3779	59.1399	3.8248E − 09
HFPSO	59.1263	15.0724	21.0304	37.1089	3.0670E − 10
CGSA10	50.3580	17.4541	14.4831	33.3329	1.3616E − 09
I-GWO	49.8187	19.4205	16.7202	43.0977	**2.7009E − 12**
ChOA	46.5509	12.0000	26.1901	47.4641	9.9216E − 10
EChOA	43.7742	19.0186	16.9050	49.1866	**2.7009E − 12**

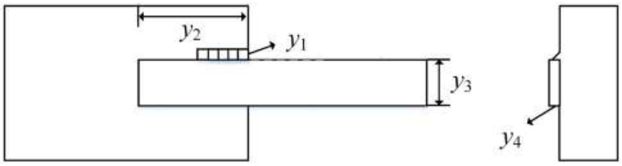

Fig. 7 Welded beam [49]

Table 15 Experimental results of welded beam design

Algorithm	y_1	y_2	y_3	y_4	F_{min}
EO	0.2057	3.4705	9.0366	0.2057	1.7249
LFD	0.1857	3.9070	9.1552	0.2051	1.7700
HFPSO	0.2035	3.5284	9.0042	0.2072	1.7354
CGSA10	0.2088	3.4205	8.9975	0.2100	1.7483
I-GWO	0.2057	3.4705	9.0366	0.2057	1.7249
ChOA	0.2134	3.3812	8.7118	0.2289	1.8382
EChOA	0.2007	3.3476	9.0415	0.2058	**1.7020**

Comparison with other algorithms on CEC2017 benchmark functions

In this subsection, 15 benchmark functions from CEC2017 [47] are used to further evaluate the performance of the proposed EChOA algorithm. The name, the class, and the global optimum of functions are given in Table 7. The number and the range of variables are all 10 and [− 100, 100], respectively. Compared with the classical benchmark functions, the selected benchmark functions with constraints and high computational cost are challenging landscapes.

Table 8 gives the average (Avg) and the standard deviation (Std) of fitness obtained by each algorithm on some CEC2017. Because these functions provide complex shapes and many local optimums, they approximate the search space of real-world problems. In terms of Avg, EChOA obtains the best results on 73.33% of functions. For "F09", I-GWO algorithm ranks first, and the proposed algorithm obtains the second result. In terms of Std, EChOA gives lower values on 66.67% of functions. These results may prove that the proposed algorithm can transition from high exploration to more exploitation. In other words, a smooth relationship is achieved between them. The convergence curve shows the trend of fitness value. Figure 5 gives convergence curves of F05, F07, F09, F25, F26, and F29. Comparing each convergence curve, EChOA can effectively find a better minimum. Effective updates are guaranteed both in the early and late iterations.

The computational complexity can qualitatively describe the runtime of an algorithm. The computational complexity of EChOA is given based on four factors such as the number of chimp N, the number of dimension D, the maximum number of iterations T, the cost of function F. In the initialization process, the computational complexity is $O(N)$. The computational complexity of function evaluation is $O(T \times N \times F)$. The computational complexity is expressed as $O(T \times N)$ for the attacker selection. The computational complexity of updating the less fit chimps' position based on beetle antennae operator is $O(T \times (N-3) \times D \times 3)$, where $D \times 3$ represents the left, right and new position of them. The computational complexity of updating the attacker's, the barrier's, and the chaser's positions is $O(T \times 3 \times D)$. The overall computational complexity of EChOA is $O(N \times (1+T \times (F+1+3D) - 6TD))$. From the quantitative analysis results, Table 9 gives the average runtime obtained by each algorithm on some CEC2017. To conclude more intuitively, the ranking of the runtime of each algorithm in most cases is as follows: LFD > EChOA > ChOA > CGSA10 > I-GWO > E O > HFPSO. From the results, it can be argued that EChOA consumes more time than ChOA and ranks second to last. An explanation is that introduced strategies are added to the

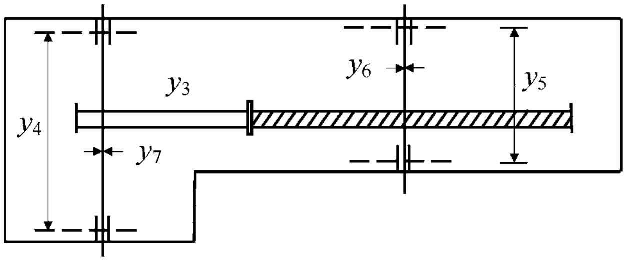

Fig. 8 Speed reducer [49]

native algorithm. The high time consumption of ChOA itself is also the main reason. To improve the accuracy of solutions, we sacrifice some runtime. It can also be concluded that the limitation of EChOA is still the computational complexity that needs to be reduced.

A full statistical analysis of results based on different performance measures is included and commented on below. Wilcoxon rank-sum test performs a rank sum test of the hypothesis that two independent samples and returns the p-value from the test. The null hypothesis is defined as no significant difference between the two samples. In this paper, the level of significance is stipulated as 0.05. $p > 0.05$ casts doubt on the validity of the null hypothesis. The other non-parametric statistical test, the Friedman test, is used to detect significant differences between the behaviors of two or more algorithms. It returns an additional structure, including ranking, statistic, and p-value [48]. Table 10 gives the exact p-values based on the Wilcoxon rank-sum test. From Table 10, it can be observed p-values are less than 0.05 for three types of functions. EChOA has significant improvements over other algorithms. Besides p-values, Table 11 also gives rankings and Chi-squares (Chi-sq) based on the Friedman test. EChOA gets the 1st rank compared to other algorithms. When the degree of freedom is 6, and the level of significance is 0.05, the critical value of the test statistic is 12.592. The calculated Chi-squared is greater than 12.592. It also proves that there are significant differences between the proposed algorithm and the other comparison algorithms.

Analysis of control parameters

In this subsection, the sensitivity of control parameters in EChOA is analyzed using experiments, as shown in Tables 12 and 13. The experimental design is consistent with the previous subsection. It can be found from the details of EChOA that the main parameters include the mutation index η_m and the attenuation rate of the step size K. The function of parameter η_m is to control the diversity of the population. The function of parameter K is to determine the step length of the search. In Table 13, the best ranking is achieved when $\eta_m = 1$ and $K = 0.95$. Besides, the results do not increase or regularly decrease with the changes of η_m and K. This is because the small value of η_m attempts to maintain a large diversity of solutions. And we need a smaller step size to obtain a more accurate optimum. But it is also a fact that too large diversity in the early phase is not beneficial to the later convergence, and too small a step size will lead to more computational consumption. So the ranking result is a trade-off between them.

Engineering optimization experiments

In this section, to highlight the applicability of the proposed EChOA algorithm, three engineering design problems are selected for further testing. They are gear train design, welded beam design, and speed reducer design. The descriptions and mathematical models of all engineering problems are provided in detail below. All issues are implemented in MATLAB through the barrier penalty function. Each algorithm runs independently for each project 30 times, with a selected chimp population of 30 and an iteration of 500. Finally, the corresponding evaluations for different projects are given.

Design of gear train

This is an unconstrained optimization problem that was proposed by Sandgren. Figure 6 shows the gear train with four variables (y_1, y_2, y_3, and y_4). The final aim is to minimize the

Table 16 Experimental results of speed reducer design

Algorithm	y_1	y_2	y_3	y_4	y_5	y_6	y_7	$F_{\min}$
EO	3.5000	0.7000	17.0000	7.3002	7.8000	3.3502	5.2867	2996.3567
LFD	3.5161	0.7000	17.0000	7.6815	7.8000	3.3513	5.2891	3007.7820
HFPSO	3.5240	0.7000	17.0000	7.6314	7.9007	3.4138	5.3834	3003.7076
CGSA10	3.5003	0.7000	17.0000	7.3350	7.8806	3.3692	5.2897	3005.5830
I-GWO	3.5000	0.7000	17.0000	7.3000	7.8000	3.3502	5.2867	2996.3482
ChOA	3.5005	0.7000	17.0000	7.3347	7.8806	3.3690	5.2899	3005.5828
EChOA	3.5000	0.7000	17.0000	7.3000	7.71532	3.3502	5.2867	**2994.4711**

gear ratio ($\frac{y_2 y_3}{y_1 y_4}$). The mathematical model is designed as follows:

$$\min F(y_1, y_2, y_3, y_4) = \left(\frac{1}{6.931} - \frac{y_2 y_3}{y_1 y_4}\right)^2 \tag{15}$$

where $12 \leq y_1, y_2, y_3, y_4 \leq 60$.

The results of variables and fitness are reported in Table 14. EChOA obtains the best fitness corresponding to the optimal solution (43.7742, 19.0186, 16.9050, 49.1866). Compared with other algorithms, EChOA can be considered to be more suitable for solving this design problem.

Design of welded beam

As it is named, this problem deals with designing a welded beam to minimize the fabrication cost. The minimization process is subject to constraints such as shear stress, bending stress in the beam, buckling load on the bar, end deflection of the beam, and side constraints. This optimum design has four parameters: the thickness of weld (y_1), the length of the clamped bar (y_2), the height of the bar (y_3), and the thickness of the bar (y_4), as shown in Fig. 7. The mathematical formulation is also illustrated as follows:

$$\min F(y_1, y_2, y_3, y_4) = 1.10471 y_1^2 y_2 + 0.04811 y_3 y_4 (14.0 + y_2) \tag{16}$$

$$\begin{aligned}
&s.t.\ \tau(y_1, y_2, y_3, y_4) - \tau_{\max} \leq 0,\ \sigma(y_1, y_2, y_3, y_4) - \sigma_{\max} \leq 0 \\
&\delta(y_1, y_2, y_3, y_4) - \delta_{\max} \leq 0,\ y_1 - y_4 \leq 0 \\
&P - P_c(y_1, y_2, y_3, y_4) \leq 0,\ 0.125 - y_1 \leq 0 \\
&1.10471 y_1^2 y_2 + 0.04811 y_3 y_4 (14.0 + y_2) - 5.0 \leq 0
\end{aligned} \tag{17}$$

where $0.1 \leq y_1, y_4 \leq 2, 0.1 \leq y_2, y_3 \leq 10$.

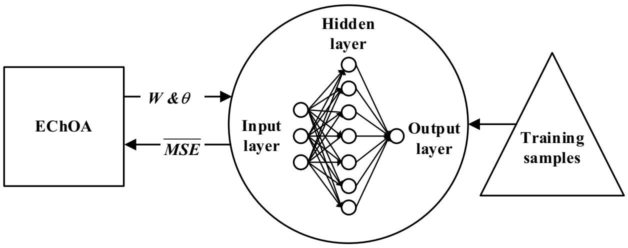

Fig. 9 EChOA for training MLP

Table 17 Descriptions of datasets

Dataset	No. of features	No. of instances	No. of classes
Balloon	4	16	2
Breast cancer	9	699	2

$$\begin{aligned}
&\tau(y_1, y_2, y_3, y_4) = \sqrt{(\tau')^2 + 2\tau'\tau''\frac{y_2}{2R} + (\tau'')^2} \\
&\tau' = \frac{P}{\sqrt{2} y_1 y_2}, \tau'' = \frac{MR}{J}, M = P\left(L + \frac{y_2}{2}\right) \\
&R = \sqrt{\frac{y_2^2}{4} + \left(\frac{y_1 + y_3}{2}\right)^2} \\
&J = 2\left\{\sqrt{2} y_1 y_2 \left[\frac{y_2^2}{4} + \left(\frac{y_1 + y_3}{2}\right)^2\right]\right\} \\
&\sigma(y_1, y_2, y_3, y_4) = \frac{6PL}{y_4 y_3^2}, \delta(y_1, y_2, y_3, y_4) = \frac{6PL^3}{E y_3^2 y_4} \\
&P_c(y_1, y_2, y_3, y_4) = \frac{4.013E\sqrt{\frac{y_3^2 y_4^6}{36}}}{L^2}\left(1 - \frac{y_3}{2L}\sqrt{\frac{E}{4G}}\right) \\
&P = 6000\ \text{lb},\ L = 14\ \text{in},\ E = 30 \times 1^6 \text{psi},\ G = 12 \times 10^6 \text{psi} \\
&\tau_{\max} = 13{,}600\ \text{psi},\ \sigma_{\max} = 30{,}000\ \text{psi},\ \delta_{\max} = 0.25\ \text{in}
\end{aligned} \tag{18}$$

Experimental results of the welded beam design problem are shown in Table 15. It is shown that the proposed algorithm can find the lowest cost design. Thus, it is reasonable

Table 18 Experimental results of different datasets

Dataset	Algorithm	$\overline{\text{MSE}} \pm$ Std	Classification rate
Balloon	EO	3.7450E − 18 ± 6.1675E − 18	**100**
	LFD	2.3629E − 10 ± 1.0864E − 09	**100**
	HFPSO	4.8369E − 20 ± 9.2670E − 19	**100**
	CGSA10	5.0816E − 11 ± 4.7952E − 10	**100**
	I-GWO	1.2257E − 25 ± 2.5536E − 25	**100**
	ChOA	3.7269E − 09 ± 5.1430E − 09	**100**
	EChOA	**4.9158E − 33 ± 1.4748E − 32**	**100**
Breast cancer	EO	3.9000E − 03 ± 4.8572E − 04	94
	LFD	5.6858E − 03 ± 6.8054E − 04	86
	HFPSO	1.2800E − 03 ± 1.8476E − 04	98
	CGSA10	4.1453E − 03 ± 5.6329E − 04	97
	I-GWO	1.2100E − 03 ± 8.7560E − 05	98
	ChOA	1.5035E − 03 ± 3.4985E − 04	98
	EChOA	**1.0300E − 03 ± 2.4868E − 05**	**99**

to think that the proposed algorithm is feasible in solving such a problem.

Design of speed reducer

This structural optimization problem involves seven variables: the face width (y_1), the module of teeth (y_2), the number of teeth on pinion (y_3), the length of shafts between bearings (y_4, y_5), and the diameter of shafts (y_6, y_7). Figure 8 shows the speed reducer. Considering nine constraints, the aim is to find the minimum weight of the speed reducer. The mathematical model is designed as follows:

$$\begin{aligned}\min F(y_1, y_2, y_3, y_4, y_5, y_6, y_7) = 0.7854y_1y_2^2(3.3333y_3^2 \\ + 14.9334y_3 - 43.0934) - 1.508y_1 \\ (y_6^2 + y_7^2) + 7.4770(y_6^3 + y_7^3) \\ + 0.7854(y_4y_6^2 + y_5y_7^2)\end{aligned} \tag{19}$$

$$\begin{aligned} & s.t.\ \frac{27}{y_1y_2^2y_3} - 1 \le 0,\ \frac{397.5}{y_1y_2^2y_3^2} - 1 \le 0 \\ & \frac{1.93}{y_2y_3y_4^3y_6^4} - 1 \le 0,\ \frac{1.93}{y_2y_3y_4^3y_7^4} - 1 \le 0 \\ & \frac{\sqrt{1.69 \times 10^6 + \left(\frac{745y_4}{y_2y_3}\right)^2}}{110y_6^3} - 1 \le 0 \\ & \frac{\sqrt{157.5 \times 10^6 + \left(\frac{745y_4}{y_2y_3}\right)^2}}{87y_7^3} - 1 \le 0 \\ & \frac{y_2y_3}{40} - 1 \le 0,\ \frac{y_1}{40} - 1 \le 0,\ \frac{5y_2}{y_1 - 1} - 1 \le 0 \end{aligned} \tag{20}$$

where $y_1, y_2, y_3, y_4, y_5, y_6, y_7 \in R$.

The results of variables and fitness are reported in Table 16. As it represents, EChOA has the superior capability to minimize the weight of the speed reducer compared to other algorithms. This example also highlights the applicability of the proposed algorithm.

Training multilayer perceptron experiments

In this section, it is the first time to use the enhanced ChOA algorithm to train multilayer perceptron (MLP). MLP is a feedforward artificial neural network model, which can be used to solve classification and regression problems. In these perceptrons, there is at least one hidden layer besides one input layer and one output layer. Data information is transmitted in one direction only. The training process is as follows.

The weighted (W) sum of inputs (x) is calculated using Eq. (21):

$$s_j = \sum_{i=1}^{n} (W_{ij}x_i - \theta_j), j = 1, 2, ..., h \tag{21}$$

where n is the number of the input nodes, h is the number of the hidden nodes, and θ represents the bias. The output of hidden nodes is calculated using Eq. (22):

$$S_j = \frac{1}{1 + e^{-s_j}}, j = 1, 2, ..., h \tag{22}$$

The final outputs are calculated based on outputs of each hidden node using Eqs. (23) and (24):

$$o_p = \sum_{j=1}^{h} W_{j,p}S_j - \theta_p^{'}, p = 1,\ 2,\ \ ...,\ m \tag{23}$$

$$O_p = \frac{1}{1 + e^{-o_p}}, p = 1, 2,\ ...,\ m \tag{24}$$

where p is the number of the output nodes. From the mathematical model, it can be analyzed that the weights W and biases θ are the most important parameters for defining the final outputs of perceptrons. The average mean square error ($\overline{MSE}$) is selected to evaluate the training model. The formula is shown as the following:

$$\overline{MSE} = \frac{\sum_{k=1}^{s} \sum_{i=1}^{m} (o_i^k - d_i^k)^2}{s} \tag{25}$$

where s is the number of the training samples, m is the number of outputs, d represents the desired output, o represents the actual output. After analysis, training MLP can be transformed into minimizing the objective function $\overline{MSE}$ by optimizing two parameters W and θ. In this paper, the Balloon dataset and Breast cancer dataset from the UCI repository [50] are used to evaluate the performance of EChOA for training MLP. Figure 9 shows the flowchart of the training process. Descriptions of these datasets are shown in Table 17. Assuming that the range of variables is [− 10, 10]. Each algorithm runs 10 times independently, with a population size of 30 and the iteration number of 250. The number of hidden nodes is equal to $2N+1$ where N indicates the number of features of datasets [51].

Results and discussions

Table 18 gives $\overline{MSE}$, standard deviation (Std), and classification rate obtained by each algorithm on two datasets. For Balloon dataset, all algorithms achieve 100% classification accuracy due to its simplicity. EChOA provides best $\overline{MSE}$

and Std. The second best results belong to I-GWO. For the Breast cancer dataset, EChOA is superior to other comparison algorithms in terms of "$\overline{MSE} \pm \text{Std}$" and classification accuracy. Each dataset has a different search space, which needs the algorithm to a higher level of the local optimum avoidance. These results also prove the merit of the proposed algorithm. EChOA can find more appropriate weights and biases for multilayer perceptrons, making the performance of the neural network better.

Conclusions and future research

In this paper, an enhanced chimp optimization algorithm named EChOA is proposed based on three strategies. HDPM enhances the diversity of the population. The correlation coefficient is calculated to determine the individuals that need to be improved. Then the beetle antennae operator helps the less fit individual to jump out of the local optimum. To explore the best performance, the strategy combinations, qualitative indicators, scalability, and control parameters of EChOA are tested. Besides, EChOA is compared with different algorithms from 2017 to 2021 based on 27 unimodal, multimodal, hybrid, and composite benchmark functions. The exploitation, exploration, and local optimum avoidance capabilities of EChOA are analyzed comprehensively. Some statistical tests with 5% confidence evaluate the significance of the improvements. Finally, three engineering design problems and training multilayer perceptron are selected as the real-world development of EChOA. The overall experimental results show that EChOA has significant competitiveness for some state-of-art algorithms.

However, like other optimization algorithms, EChOA also has some limitations that need to be improved. By analyzing the computational complexity and runtime in the experiment section, high consumption is still the main limitation. We believe that this problem can be alleviated by introducing some parallel strategies such as the island model or co-evolutionary mechanism.

For future research, attention will be given to how to reduce the process cost without loss of accuracy of the solution. In this paper, EChOA resolved the parameters tuning problem with the multilayer perceptron. In the next case, the development of a binary version to evaluate the potential features from the pool of features of a given machine learning/deep learning problem is an exciting topic. Other applications, including image segmentation, Internet of Things, and multi-objective optimization are also worth investigating.

Acknowledgements This work was supported by Sanming University introduces high-level talents to start scientific research funding support Project (20YG14), Guiding science and technology projects in Sanming City (2020-G-61), Educational research projects of young and middle-aged teachers in Fujian Province (JAT200618), Scientific research and development fund of Sanming University (B202009).

Declarations

Conflict of interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

References

1. Hubner LG, Maletzke AG, de Nadai BL, Schaefer RL, Zalewski W, Ferrero CA (2017) FB-DT: an improvement in the Brute Force algorithm for motifs discovery. IEEE Lat Am Trans 15(8):1542–1546
2. Alzaqebah M, Abdullah S (2014) An adaptive artificial bee colony and late-acceptance hill-climbing algorithm for examination timetabling. J Sched 17:249–262
3. Eshragh A, Filar JA, Haythorpe M (2011) A hybrid simulation-optimization algorithm for the Hamiltonian cycle problem. Ann Oper Res 189:103–125
4. Alipour MM, Razavi SN, Feizi Derakhshi MR, Balafar MA (2018) A hybrid algorithm using a genetic algorithm and multiagent reinforcement learning heuristic to solve the traveling salesman problem. Neural Comput Applic 30:2935–2951
5. Porumbel DC (2012) Heuristic algorithms and learning techniques: applications to the graph coloring problem. 4OR-Q J Oper Res 10:393–394
6. Gandomi AH, Yang X-S, Talatahari S, Alavi AH (2013) Metaheuristic algorithms in modeling and optimization. Metaheuristic applications in structures and infrastructures. Elsevier, Oxford, pp 1–24
7. Sun K, Jia H, Li Y, Jiang Z (2021) Hybrid improved slime mould algorithm with adaptive β hill climbing for numerical optimization. J Intell Fuzzy Sys 40(1):1667–1679
8. Hussain A, Muhammad YS (2020) Trade-off between exploration and exploitation with genetic algorithm using a novel selection operator. Complex Intell Syst 6:1–14
9. Rashedi E, Nezamabadi-pour H, Saryazdi S (2009) GSA: a gravitational search algorithm. Inf Sci 179(13):2232–2248
10. Mirjalili S, Mirjalili SM, Hatamlou A (2016) Multi-verse optimizer: a nature-inspired algorithm for global optimization. Neural Comput Appl 27:495–513
11. Kaveh A, Dadras A (2017) A novel meta-heuristic optimization algorithm: thermal exchange optimization. Adv Eng Softw 110:69–84
12. Corriveau G, Guilbault R, Tahan A, Sabourin R (2016) Bayesian network as an adaptive parameter setting approach for genetic algorithms. Complex Intell Syst 2:1–22

13. Storn R, Price K (1997) Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces. J Global Optim 11(4):341–359
14. Kennedy J, Eberhart R (1995) Particle swarm optimization. In: Proceedings of ICNN'95-International Conference on Neural Networks, pp. 1942–1948. IEEE, Perth
15. Mirjalili S (2015) The ant lion optimizer. Adv Eng Softw 83:80–98
16. Mirjalili S, Lewis A (2016) The whale optimization algorithm. Adv Eng Softw 95:51–67
17. Faramarzi A, Heidarinejad M, Mirjalili S, Gandomi AH (2020) Marine predators algorithm: a nature-inspired metaheuristic. Expert Syst Appl 152:113377
18. Rao RV, Savsani VJ, Vakharia DP (2011) Teaching-learning-based optimization: a novel method for con-strained mechanical design optimization problems. Comput-Aided Des 43(3):303–315
19. Ho YC, Pepyne DL (2020) Simple explanation of the no-free-lunch theorem and its implications. J Optim Theory Appl 115:549–570
20. Sörensen K (2015) Metaheuristics—the metaphor exposed. Int Tran Oper Res 22(1):3–18
21. Michalewicz Z (2012) Quo vadis, evolutionary computation?: on a growing gap between theory and practice. In: Proceedings of the 2012 World Congress conference on Advances in Computational Intelligence, pp. 98–121. Springer, Brisbane
22. Khishe M, Mosavi MR (2020) Classification of underwater acoustical dataset using neural network trained by chimp optimization algorithm. Appl Acoust 157:107005
23. Arab A, Alfi A (2015) An adaptive gradient descent-based local search in memetic algorithm applied to optimal controller design. Inf Sci 299:117–142
24. Li Z, Wang W, Yan Y, Li Z (2015) PS-ABC: a hybrid algorithm based on particle swarm and artificial bee colony for high-dimensional optimization problems. Expert Syst Appl 42(22):8881–8895
25. Berkan Aydilek İ (2018) A hybrid firefly and particle swarm optimization algorithm for computationally expensive numerical problems. Appl Softw Comput 66:232–249
26. Mirjalili S, Gandomi AH (2017) Chaotic gravitational constants for the gravitational search algorithm. Appl Softw Comput 53:407–419
27. Khishe M, Mosavi MR, Moridi A (2018) Chaotic fractal walk trainer for sonar data set classification using multi-layer perceptron neural network and its hardware implementation. Appl Acoust 137:121–139
28. Kaveh M, Khishe M, Mosavi MR (2019) Design and implementation of a neighborhood search biogeography-based optimization trainer for classifying sonar dataset using multi-layer perceptron neural network. Analog Integr Circ Sig Process 100:405–428
29. Jia H, Sun K, Song W, Peng X, Lang C, Li Y (2019) Multi-strategy emperor penguin optimizer for RGB histogram-based color satellite image segmentation using Masi entropy. IEEE Access 7:134448–134474
30. Khishe M, Mosavi MR (2019) Improved whale trainer for sonar datasets classification using neural network. Appl Acoust 154:176–192
31. Khishe M, Mohammadi H (2019) Passive sonar target classification using multi-layer perceptron trained by salp swarm algorithm. Ocean Eng 181:98–108
32. Mousavi Y, Alfi A, Kucukdemiral IB (2020) Enhanced fractional chaotic whale optimization algorithm for parameter identification of isolated wind-diesel power systems. IEEE Access 8:140862–140875
33. Shokri-Ghaleh H, Alfi A, Ebadollahi S, Shahri AM, Ranjbaran S (2020) Unequal limit cuckoo optimization algorithm applied for optimal design of nonlinear field calibration problem of a triaxial accelerometer. Meas 164:107963
34. Ma S, Li D, Hu T, Xing Y, Yang Z, Nai W (2020) Huber loss function based on variable step beetle antennae search algorithm with Gaussian direction. In: 12th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC), pp. 248–251. IEEE, Hangzhou
35. Wang H, Wang W, Zhou X, Zhao J, Wang Y, Xiao S, Xu M (2020) Artificial bee colony algorithm based on knowledge fusion. Complex Intell Syst. https://doi.org/10.1007/s40747-020-00171-2
36. Nadimi-Shahraki MH, Taghian S, Mirjalili S (2021) An improved grey wolf optimizer for solving engineering problems. Expert Syst Appl 166:113917
37. Khishe M, Mosavi MR (2020) Chimp optimization algorithm. Expert Syst Appl 149:113338
38. Abed-Alguni B (2019) Island-based cuckoo search with highly disruptive polynomial mutation. Int J Artif Intell 17(1):57–82
39. Deb K, Tiwari S (2008) Omni-optimizer: a generic evolutionary algorithm for single and multi-objective optimization. Eur J Oper Res 185(3):1062–1087
40. Dhargupta S, Ghosh M, Mirjalili S, Sarkar R (2020) Selective opposition based grey wolf optimization. Expert Syst Appl 151:113389
41. Jiang X, Li S (2017) BAS: beetle antennae search algorithm for optimization problems. Int J Rob Control 1(1):1
42. Khan AH, Li S, Luo X (2020) Obstacle avoidance and tracking control of redundant robotic manipulator: an RNN-based metaheuristic approach. IEEE Trans Ind Inf 16(7):4670–4680
43. Available: https://wikimili.com/en/Antenna_(biology)#Rosalia_alpina_side.JPG
44. Faramarzi A, Heidarinejad M, Stephens B, Mirjalili S (2020) Equilibrium optimizer: a novel optimization algorithm. Knowl Based Syst 191:105190
45. Houssein EH, Saad MR, Hashim FA, Shaban H, Hassaballah M (2020) Lévy flight distribution: a new metaheuristic algorithm for solving engineering optimization problems. Eng Appl Artif Intell 94:103731
46. Digalakis JG, Margaritis KG (2001) On benchmarking functions for genetic algorithms. Int J Comput Math 77(4):481–506
47. Awad NH, Ali MZ, Suganthan PN (2017) Ensemble sinusoidal differential covariance matrix adaptation with Euclidean neighborhood for solving CEC2017 benchmark problems. In: 2017 IEEE Congress on Evolutionary Computation (CEC), pp. 372–379. IEEE, San Sebastian
48. Carrasco J, García S, Rueda MM, Das S, Herrera F (2020) Recent trends in the use of statistical tests for comparing swarm and evolutionary computing algorithms: practical guidelines and a critical review. Swarm Evol Comput 54:100665
49. Gandomi AH, Yang XS, Alavi AH (2013) Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems. Eng comput 29:17–35
50. Dua D, Graff C (2019) UCI machine learning repository. Available: http://archive.ics.uci.edu/ml
51. Mirjalili S (2015) How effective is the grey wolf optimizer in training multi-layer perceptrons. Appl Intell 43(1):150–161