Improved Dandelion Optimizer Based on Elite Pool and Mirror Optimization Strategy

Zhenlong Zhao, Zhongfeng Li*, Tiefeng An and Shiqi Wei

Abstract—**A new swarm intelligence optimization algorithm is proposed to simulate the long-distance flight process of dandelion seeds as a function of wind by the dandelion optimizer (DO). An improved version of DO, called IDO, is proposed to overcome the shortcomings of slow optimization speed and vulnerability to local optimum of the algorithm. In the landing stage of the dandelion seeds, IDO introduces the elite pool roulette wheel selection mechanism to maintain the diversity of the dandelion population, which enables the algorithm to maintain a better optimization ability. In the rising stage, a gamma distribution is introduced to make the algorithm population more diverse, thus improving its optimization capabilities. Finally, the mirror reversal strategy is introduced to search for the mirror solutions of inferior particles in the population surrounding the optimal individual, which helps the population to escape from local optima. The performance of the proposed improved IDO algorithm was tested against the CEC-2017, CEC-2020, and CEC-2022 benchmark functions. Finally, three engineering design problems were optimized, namely, three-bar truss design, cantilever design, and pressure vessel design. The experimental results show that the proposed IDO algorithm significantly improves the convergence rate and optimization accuracy. The experimental results show that the proposed IDO algorithm optimizes the exploitation and exploration process, improves the convergence rate and optimization accuracy.**

Index Terms—**Dandelion optimizer, Elite pool, Mirror Optimization Strategy, function optimization**

I. Introduction

THE Times is growing quickly, and people are producing and living richer lives. The complexity of numerical optimization problems and the difficulty of real-world optimization problems both increase, necessitating the use of effective methods to solve them [1]. Modern optimization problems cannotd always be solved using the conventional optimization techniques. More reliable optimization strategies are desperately needed [2] due to the problem's rising complexity and difficulty. Identification of decision factors while preserving a variety of restrictions to maximize or decrease the cost function is known as optimization. Any optimization problem must have constraints, cost functions, and design variables. To complete the system design at the lowest possible cost, the optimization procedure entails determining the ideal value for the unique system parameters [3]. Among Engineering [4]-[5], feature selection [6]-[8], image processing [9]-[10], wireless sensor networks [11]-[12], machine learning parameter optimization [13]-[15], medical [16]-[17], agricultural [18]-[19], finance [20]-[21], and other fields all make extensive use of optimization techniques. Many problems encountered in daily life are generally non-convex and nonlinear due to the intrinsic nature of many design components and restrictions. Furthermore, there is no guarantee that an all-encompassing optimum answer will be discovered. Scientists develop novel solutions to these real-world issues to produce better results.

Metaheuristic algorithms are widely used and have high potential due to their wide applicability in various techniques based on randomized methods. The main purpose of introducing metaheuristic algorithms is to better solve optimization problems and achieve better results [22]. Therefore, some researchers not only try to propose new metaheuristic algorithms, but also try to improve the efficiency of existing methods. There are some successful metaheuristic algorithms such as particle swarm optimization (PSO) [23], ant lion algorithm (ALO) [24], grey wolf optimizer (GWO) [25], marine predator algorithm (MPA) [26], etc. Metaheuristic algorithms can be broadly divided into three types: (1) Evolutionary algorithms [27], including mainly GA [25], DE [28], etc. (2) Swarm intelligence-based algorithm (SI) is characterized by self-organization and collective intelligence behavior of decentralized systems. Bonabeau defines SI as "sudden collective intelligence of a simple group of agents" [29]. (3) Algorithms based on representation of human behavior [30] are mainly inspired by human behavior. Better solutions can be generated by human behavior and other means until the final standards are met, such as the teaching-learning-based optimization algorithm (TLBO) [31], the equilibrium optimizer (EO) [32], and the social group optimization algorithm (SGO) [33]. It should be emphasized that the existing metaheuristic algorithms have advantages and limitations. The coordinate between exploration and evolution is crucial for metaheuristic algorithms all the time [34]. The algorithm that maintains this coordinate in different optimization problems is a successful algorithm.

In recent years, new algorithms have gradually appeared, and many new optimization algorithms have been used to solve problems such as technical optimization. Shabani proposed the search and rescue optimization algorithm (SAR)

Manuscript received Aug 31, 2023; revised Dec 8, 2023.

This work was supported in part by the Scientific research fund project of Liaoning Provincial Department of Education (Grant No. LJKMZ20221859 Grant No. LJKZ1199, Grant No.2020-YKLH-32). 2022 enterprise and doctor double innovation plan project of Yingkou City (Grant No. QB-2022-04).

Zhenlong Zhao is a Senior Engineer in School of Electrical Engineering, Yingkou Institute of Technology, Yingkou, 115014, P. R. China (E-mail:zzl_idea@126.com)

Zhongfeng Li is an Associate Professor in School of Electrical Engineering, Yingkou Institute of Technology, Yingkou, 115014, P. R. China (corresponding author to provide E-mail: alyx0640@163.com)

Tiefeng An is a Senior Engineer in the Steelworks of Beiying Iron and Steel Group Co.,Ltd of Benxi Steel Group Corporation, Liaoning, Benxi, 117000, P. R. China (phone: 86-13842446121, E-mail: tiefengan@126.com)

Shiqi Wei is an Engineer of Yokogawa China Co., Ltd., Shanxi, Xi'an, 710000, PR China (E-mail: shiqi100@126. com)

[35]. He proposes a restart strategy to avoid locally infeasible minimum values in some complex constrained optimization problems. Hayyolalam et al. proposed a new metaheuristic algorithm for continuous nonlinear optimization problems, the black widow optimization algorithm (BWO) [36]. This algorithm was inspired by the unique mating behavior of latrodectus mactans. The experimental results show that the algorithm has advantages in early convergence and fitness optimization. Eskandar et al. proposed the water cycle algorithm (WCA) based on observations of natural water cycle processes and how rivers and streams flow into the ocean [37]. Pan et al. proposed the gannet optimization algorithm (GOA) based on gannet foraging [38]. GOA has two diving modes, namely the U-shaped mode and the V-shaped mode, with the goal of exploring the optimal area in the search space. Sudden turns and random walks can help the algorithm to find higher quality solutions in the corresponding areas. The artificial rabbit optimization algorithm (ARO) [39] was developed based on the subsist methods of rabbits in habitat, including finding detours and random hiding places. Hongyu Long et al. proposed a multi-strategy improved Aquila Optimization (IAO) algorithm for solving ORPD [40]. Gonggui Chen et al. proposed considering performance index and an improved Gravitational Search Algorithm (IGSA) for optimizing parameters of a fuzzy PID (FPID) controller[41].

In 2022, Zhao et al. proposed the dandelion optimizer (DO) [42]. DO replicates the long-distance flight of dandelion seeds using wind, which is divided into three phases: ascending, descending, and landing. The DO method, which is widely used to solve various optimization problems, is a powerful optimizer with many iterative optimization and strong robustness. Abbassi et al. employed the dandelion optimizer to accurately estimate the essential parameters of the PEMFC model and identified the parameters of the PEMFC model accurately for the first time [43]. The results show that the proposed strategy provided satisfactory results and outperformed recognized competing methods. Kaveh et al. proposed an improved version of DO. The improved Dandelion Optimizer (EDO) is used for steel frame construction, and the statistical regeneration mechanism (SRM) is used in EDO [44]. Halasa et al. proposed a novel DO -based technology for the implementation of GMPP. The proposed technology aims to improve the efficiency of power generation in solar systems, especially under the conditions of PS. Simulation results show that the MPPT method used has advantages in multiple evaluation dimensions such as tracking efficiency [45]. The CIDO approach was proposed by Akyol et al. [46] for global optimization of chaotic initialization DO. For the first time, chaotic initialization is added to DO. The experimental results show that the chaotic DO initialization leads to successful results.

Considering the previous applications, DO can still be utilized to address many complicated problems in the real world. However, in other cases, they may slip into local optima and lose their optimization capacity, necessitating simplification and increased robustness. To address this issue, this work introduces the improved dandelion optimizer (IDO), a new and effective DO version. The results indicate the algorithm's applicability and efficacy in optimization. In summary, this work's contribution can be summarized as follows:

- An adaptive elite pool strategy was proposed to guide population evolution and help the population emerge from local optima.
- Replace the random number in the original algorithm with a gamma distribution to improve the search step size.
- Propose a mirror optimization strategy to improve the location update of underperforming agents.
- The efficiency and performance of the proposed IDO algorithm were evaluated in the benchmark functions of CEC2017, CEC2020, and CEC2022.
- To verify the performance of the IDO algorithm, it was compared with the original algorithms DO, AOA, SCA, WOA, AVOA, MFO and HHO.
- The Friedman statistical tests were performed to confirm the statistical advantages of the proposed IDO.
- The convergence graph analysis shows the consistency of the proposed IDO algorithm.
- Evaluate the solving ability of the proposed IDO algorithm by solving four practical engineering design problems (three-bar truss design problem, pressure vessel problem, welded beam problem).

The remainder of this paper is organized as follows: Section 2 presents the mathematical model, working principle, and pseudocode of DO. Section 3 introduces the adaptive elite pool strategy, gamma distribution mathematical model, and mirror update strategy used in this study, and provides a flowchart of IDO. In Section 4, the proposed IDO algorithm is applied to three different sets of benchmark functions and three different real-world engineering design problems. Section 5 provides an overview of the conclusions of this study and the work that remains to be done.

II. Basic Principles of Dandelion Optimizer

The mathematical expression for DO is described in detail in this section.

A. Initialize

Like most metaheuristic algorithms, DO also has an initial population. Assuming that each dandelion seed represents a potential solution in the problem search space, the entire dandelion seeds population can be described by (1)

$$population = \begin{bmatrix} x_1^1 & \dots & x_1^{Dim} \\ \vdots & \ddots & \vdots \\ x_{pop}^1 & \dots & x_{pop}^{Dim} \end{bmatrix} \tag{1}$$

where pop represents the proxy size and Dim represents the size of search space. Each agent is created between the upper bound (UB) and the lower bound (LB) of the solution space at random, X_i represents the ith individual, whose expression (2) presents.

$$X_i = \text{rand} * (\text{UB} - \text{LB}) \quad \text{LB} \tag{2}$$

Where *rand* is the random number between [0,1], *Lb* and *UB* are represented by (3)

$$LB = [lb_1, \dots, lb_{Dim}]$$

$$UB = [ub_1, \ldots, ub_{Dim}] \tag{3}$$

The algorithm DO uses the agent with the best fitness value as the elite to guide evolution during the initialization process, which can be considered the most appropriate location for dandelion seed growth. The mathematical expression for the initial elite X_{elite} can be expressed as follows.

$$\begin{gathered} f_{\text{best}} = \min\left(f(X_i)\right) \\ X_{\text{elite}} = X(\text{find}(f_{\text{best}} == f(X_i))) \end{gathered} \tag{4}$$

where $find(f_{\text{best}} == f(X_i))$ represents an index where two values are equal.

B. Rising Stage

The ascending phase is critical to the dispersal of dandelion seeds, which reach a certain height before they can fly away from their parents. How high the seeds rise depends on several factors, including wind speed, humidity, and other factors. Based on these factors, weather can be divided into two categories.

Case1: On clear days, wind speed can be considered lognormal $\ln Y \sim N(\mu, \sigma^2)$. In this case, since the seeds are mainly distributed along the Y-axis, the seed propagation is remote and random, which triggers the DO exploration process. Within the search area, the dispersal of dandelion seeds is highly dependent on wind speed, which affects both the height and distance they travel. Under this influence, the vortex above the seed is constantly adjusted, which forces the seed to spiral upward, and the corresponding formula is as follows.

$$X_{t+1} = X_t + \alpha \cdot v_x \cdot \ln Y \cdot (X_s - X_t) \tag{5}$$

Where X_t represents the location of the population during the t iteration. X_s represents a selected position at random during the s iteration. X_s is shown in (6).

$$X_s = \text{rand}(1, \text{Dim}) \cdot (\text{UB} - \text{LB}) \quad \text{LB} \tag{6}$$

Refer to (7) provides the expression for lnY, lnY follows the lognormal distribution with $\mu = 0$ and $\sigma^2 = 1$, and y is the standard normal distribution $N(0, 1)$.

$$lnY = \begin{cases} \dfrac{1}{y\sqrt{2\pi}} \exp\left(-\dfrac{1}{2\sigma^2}(\ln y)^2\right) & y \geq 0 \\ 0 \quad y < 0 \end{cases} \tag{7}$$

Adaptive parameters α is used for representation, and (8) is a mathematical expression.

$$\alpha = \text{rand} \cdot \left(\frac{1}{T^2} t^2 - \frac{2}{T} t \quad 1\right) \tag{8}$$

Fig.1(a) shows the dynamically changing of α with the number of iterations. As can be seen from Fig.1(a), the non-linear of α decreases as the number of iterations increases and gradually approaches 0, with random fluctuations between [0,1]. This type of volatility causes the algorithm to pay more attention to the global optimization in the initial phase and to move from the global search to the local search in the later phase, resulting in accurate convergence after the global optimization. v_x and v_y represent the coefficients of the dandelion's buoyancy component when exposed to the separation vortex. According to the (9), the magnitude of the force can be calculated in different orders of magnitude.

$$\begin{gathered} r = \frac{1}{e^{\theta}} \\ v_x = r \cdot \cos(\theta) \\ v_y = r \cdot \sin(\theta) \end{gathered} \tag{9}$$

Where θ is the random number between $[-\pi, \pi]$.

Case2: On rainy days, dandelion seeds are affected by air dampness and other influencing factors and cannot rise with the wind normally. Therefore, in this phase, the population is exploited in their close environment. The relevant mathematical expression can be expressed as follows.

$$X_{t+1} = X_t \cdot k \tag{10}$$

The role of k is to restrict the local search range, which is calculated using (11).

$$\begin{gathered} k = \frac{1}{T^2 - 2T + 1} t^2 - \frac{1}{-\ T^2 +\ 2T \quad 1} t \quad 1 \\ k = 1 - \text{rand}() \cdot q \end{gathered} \tag{11}$$

The value of the k gradually closes in on 1 during iteration to ensure that the dandelion seeds population finds the optimal location. To sum up, in the ascending phase of the algorithm, its mathematical representation can be expressed as.

$$X_{t+1} = \begin{cases} X_t + \alpha \cdot v_x \cdot \ln Y \cdot (X_s - X_t), & \text{randn} \quad 1.5 \\ X_t \cdot k, & \text{else} \end{cases} \tag{12}$$

where *randn* is a random number that accord with the standard normal distribution.

C. Descending Stage

At the current stage, the DO still emphasizes exploration. In DO, Brownian motion is used to simulate the movement of dandelion seeds, which simulates the decline process of dandelion seeds when they rise to a certain height. Because Brownian motion obeys a normal distribution, the agent has a wider search space. To ensure the stability of dandelion seeds in the process of falling, the seeds use the average position information in the process of population rising to guide the population to a more promising position. The mathematical expression of this process is.

$$X_{t+1} = X_t - \alpha \cdot \beta_t \cdot (X_{\text{mean}_t} - \alpha \cdot \beta_t \cdot X_t) \tag{13}$$

where β_t represents Brownian motion and is a random number in the standard normal distribution. X_{mean_t} represents the average position of the population in t iterations, and its mathematical expression is.

$$X_{\text{mean}_t} = \frac{1}{\text{pop}} \sum_{i=1}^{\text{pop}} X_i \tag{14}$$

D. Landing Stage

In the development stage, dandelion seeds select the landing site according to the results of the ascending and descending stages, and gradually reaches the global optimal with the progress of iteration, as shown in (15).

$$X_{t+1} = X_{\text{elite}} + \text{levy}(\lambda) \cdot \alpha \cdot (X_{\text{elite}} - X_t \cdot \delta) \tag{15}$$

Where X_{elite} represents the optimal position of dandelion seed in t iterations. $levy(\lambda)$ represents the Levy flight function, calculated using (16) (Mantegna, 1994).

$$\text{levy}(\lambda) = \frac{s \cdot (\omega \cdot \sigma)}{|t|^{\frac{1}{\beta}}} \tag{16}$$

where, $\beta = 1.5$, $s = 0.01$, ω and t are the random number between [0,1], σ and δ mathematical expressions for are (17) and (18), respectively.

$$\sigma = \frac{\Gamma(1+\beta) \ \sin\left(\frac{\pi\beta}{2}\right)}{\Gamma\left(\frac{1+\beta}{2}\right) \cdot \beta \cdot 2^{\frac{\beta-1}{2}}} \tag{17}$$

$$\delta = \frac{2t}{T} \tag{18}$$

The pseudo-code of the proposed DO algorithm is described in detail in Algorithm 1.

Dandelion Optimizer

Input: *Pop*, *MaxIter* and *Dim*
Output: The optimal dandelion seeds X_{best} and its fitness value f_{best}
Initialize DO population X
Calculate the fitness value f of individual.
Select the X_{elite} according to f
while ($t < MaxIter$) do
  /***Rise stage***/
  if $randn < 1.5$ do
    Generate a using (8)
    Update population X using (5)
  else if do
    Generate k using (11)
    Update population X using (10)
  end if
  /***Decline stage***/
  Update population X using (13)
  /***Land stage***/
  Update population X using (15)
  Arrange DO individual according to f
  Update X_{elite}
  if $f(X_{best}) < f(X_{elite})$
    $X_{best} = X_{elite}$, $f_{best} = f(X_{elite})$
  end if
end while
Return X_{best} and f_{best}

III. Improved Dandelion Optimizer

To ameliorate the problem of premature convergence and fall into local optima in DO, three improvement methods are proposed. First, an adaptive elite pool strategy was proposed to guide the population evolution and help the population get out of local optima; second, the random numbers in the original algorithm are replaced by a gamma distribution to improve the search step size; finally, a mirrored optimization strategy is proposed to improve the location update of underperforming agents.

A. Constructing Elite Pool Based on Wheel Reverse Selection Strategy

From the perspective of the optimization process for the algorithm, in the descending phase of each iteration, all dandelion seeds are headed by the individual with the smallest fitness value in the agent, which makes the algorithm clearer and allows for fast convergence, but also causes the algorithm to be inefficient in search and easily fall into the local optimum. Therefore, the "falling phase" search process in the dandelion optimizer is considered from the perspective of probability. Specifically, the elite pool consists of four dandelion seeds with the smallest fitness in the dandelion population, and a roulette reverse selection strategy based on fitness is proposed to select an elite individual in the elite pool to guide population evolution according to probability.

The traditional roulette strategy of using the ratio of fitness values to total fitness values as the selection probability is a greedy selection method, and this selection strategy causes the DO population to cluster in foods with high fitness values, thus reducing the diversity of the population. Therefore, the selection strategy of reverse roulette is applied into DO. The reverse roulette selection strategy is to combine the probability formula (19) in the traditional roulette strategy.

$$P_i = \frac{\text{fit}_i}{\sum_{i=1}^{N} \text{fit}_i} \tag{19}$$

Where fit_i is the fitness value of the ith solution and N is the number of solutions.

Replace with the following (20).

$$P_i = \frac{1}{\text{fit}_i} \Big/ \sum_{i=1}^{N} \frac{1}{\text{fit}_i} \tag{20}$$

That is, the ratio between the reciprocal of the fitness value and the reciprocal of the total fitness value is used to optimize the dandelion seed population. It can be seen from the formula that the greater the fitness value, the smaller the probability of the reciprocal fitness value, which can maintain the diversity of dandelion seed population, and not easily fall into the local optimal.

B. Gamma Distribution

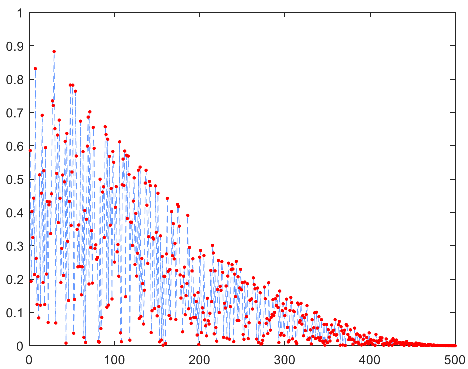
In order to further improve the DO exploitation and exploration ability, the normal distribution in the DO rise process is replaced by gamma distribution. Through reasonable parameter Settings, the gamma distribution has a large disturbance range, so as to improve the global exploration ability in the early stage and local exploitation ability in the later stage of the algorithm. The mathematical expression for the gamma distribution is as (21).

$$f(x) = \begin{cases} \dfrac{\beta^{\alpha} x^{\alpha-1} e^{-\beta x}}{\Gamma(\alpha)}, & \text{if } x > 0 \\ 0, & \text{otherwise} \end{cases} \tag{21}$$

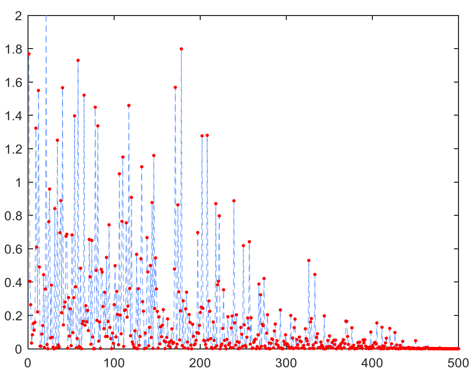
where, the mean of the gamma distribution $\mu = \dfrac{\alpha}{\beta}$,

variance $\sigma^2 = \dfrac{\alpha}{\beta}$, $\alpha = 1$, $\beta = 1$.

It can be intuitively seen in the comparison Fig. 1(a) that in the early iteration period, the number of iterations are small, and the gamma distribution has a large fluctuation range than the normal distribution. Compared with the original α, it is a fixed value related to the number of iterations, which is equivalent to the operation of the whole population in equal proportion, which is not conducive to maintaining the diversity of the population. To solve this problem, gamma distribution variation vector is introduced. From the observation of Fig.1(a), it can be seen that in the early stage of iteration, the variation step size is not obviously constrained, and a large number of its values are distributed between 0 and 0.2, while there are also a large number of values greater than 1, which means that only some positions in the agent are significantly varied, and the global exploration ability of the algorithm is enhanced while maintaining the advantages of the algorithm itself. In the late iteration, the distribution rapidly approaches 0, which enhances the local exploitation capability of DO.



(a) Original dynamic trend of α



(b) Dynamic trend of α with gamma distribution

Fig. 1. Dynamic trend of α

C. Mirror Optimization Strategy of Underperforming Dandelion Seed

Since the dandelion optimizer has a fast convergence speed, it is easy for the optimization results to fall into local optimality for a complex search space. Therefore, the mirror optimization strategy improves the dandelion population optimization, some agents search for the mirror solution about the best individual, the dandelion seed population is sorted according to fitness, and the number of underperforming agents to be updated is selected according to (22)

$$N_w = N\left(\text{rand} \cdot (c_1 - c_2) \quad c_1\right) \tag{22}$$

c_1 is 0.3 and c_2 is 0.5. After the DO position update, the mirror foraging operation is performed, the mathematical model of which is shown in (23).

$$X_i^d(t+1) = X_i^d + S(r_1 \cdot X_{\text{best}}^d - r_2 \cdot X_i^d(t)), \quad i \quad 2,3,\ldots,NP \tag{23}$$

where, S is the mirror factor, which determines the flipping range of dandelion seeds, where $S = 1$; r_1 and r_2 are uniformly distributed random numbers in the interval [0, 1]. By introducing the idea of mirror optimization into DO, the agents in the local extremum region can be released to search and optimize in a wider space.

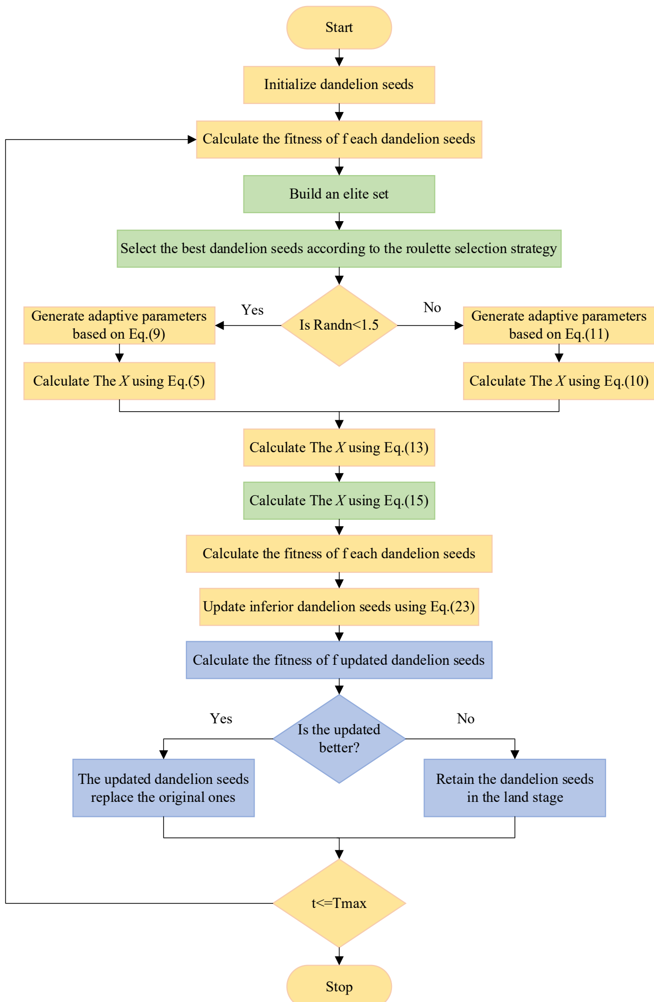


Fig. 2. Flowchart of proposed IDO

IV. Experimental results and discussions

To verify the optimization performance of the algorithm IDO, we propose several benchmark test functions, including CEC2017, CEC2020, and CEC2022 test functions, and some practical problems for evaluating the performance of the algorithm.

A. Benchmark test functions and algorithms for comparison

To test the performance of IDO, three sets of systematic and complex benchmark test function steps were used in this experiment, and three engineering optimization problems were used for testing IDO's ability to solve real-world problems. The results were also compared with some well-known optimization algorithms, including AOA [47], SCA [48], WOA [49], AVOA [50], MFO [51], HHO [52]. All experiments were performed using Matlab 2022b on Intel Core i7 computers, CPU 2.10 GHz, RAM 16GB, and Windows 11 operating systems. Table I explains the different parameters of the comparison algorithm. The parameters used are either accurately recommended by their developers or are within the accepted range of optimal performance for all algorithms.

TABLE I
ALGORITHM PARAMETER SETTINGS

Algorithm	Parameter	Value
AOA	α, μ	5, 0.5
SCA	α	2
WOA	r, l	[0,1], [-1,1]
AVOA	$P_1, P_2, P_3, W, L_1, L_2$	0.6,0.4,0.6, 2.5,0.8,0.2
MFO	a, b	[-2, -1], 1
HHO	E_0	[-1, 1]
DO	α, k	[0,1], [0,1]
IDO	α, k, c_1, c_2, S	[0,1], [0,1], 0.3, 0.5, 1

B. The performance of the IDO on the CEC2017

Description of CEC2017

To evaluate the ability of IDO to explore and avoid local minima, CEC2017 includes 30 sophisticated combined and mixed functions. The unimodal function is F1-F3, and there is only one global optimal solution in the solution space, which is used to test convergence speed and search accuracy. The multi-peaked function (F4-F10) is characterized by multiple local optima and is used to test the ability of the algorithm to jump out of the local optimum. The mixed function is F11-F20, and the composite function is F21-F30. The properties of the functions are listed in Table II. IDO tests these functions and compares them with other established methods. In this suite, the dimension of the function is set to 10. Each algorithm is run 30 times (500 iterations), and the results of each run are reported as the mean (Ave) and standard deviation (Std) of the best solution found so far.

Benchmark test functions and algorithms for comparison

Fig. 3 shows the convergence plot of the representative function, and Table III contains the average and variance of 30 runs with 30 test functions. The optimal results of the algorithm are shown in bold. From Table III and Fig. 3, it can be seen that IDO provides acceptable convergence rate and convergence accuracy for most functions compared to the standard DO. From the results, it can be seen that for unimodal function problems, the convergence rate and convergence accuracy of IDO are not as good as those of the original DO. Although the elite pool leadership optimization strategy helps the algorithm to strengthen its evolvability, it has certain disadvantages in evolvability accuracy. The multimodal function IDO performed the best in all the test sets except F4, and from the observation of the convergence graph, IDO showed good development and exploration abilities throughout the process. Compared with multimodal test functions, mixed functions are more complex and can simulate real optimization problems more accurately. The mixed function has multiple local optimal solutions. As shown in Table III, IDO will effectively improve the optimization of most of the mixed functions, and the mean value obtained on F11, F14, F15, F16, F17, F19 and F20 is the smallest. The combination functions are more complex than previously introduced functions. The minimum values were obtained for the mean values of F21, F23, F26, F28, F29, and F30, and the effect of IDO was lower than that of DO only for F25. From Fig. 3, it can be seen that IDO can maintain good development and exploration capabilities throughout the search process compared to other algorithms. However, due to the complexity of composite functions, there are still some functions (F22, etc.) that struggle to escape local optima, which leads to bad results.

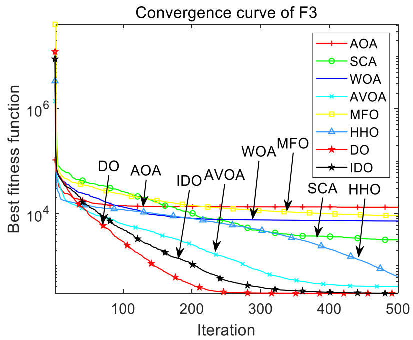
According to the Friedman average ranking, the CEC2017 test function ranks IDO first and the standard DO second. The results show that IDO is the optimum among all given optimization methods. The improved algorithm obviously improves the ability of IDO to jump out of the local optimal, avoids falling into the local optimal, and effectively improves search efficiency.

TABLE II
PROPERTIES AND SUMMARY OF THE CEC2017 TEST FUNCTIONS

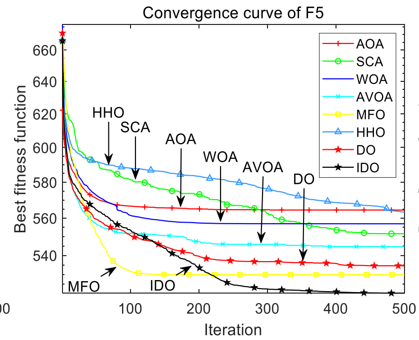
Func	No.	Dim	Range	f_{min}
Unimodal Function	1	10	[-100,100]	100
	2	10	[-100,100]	200
	3	10	[-100,100]	300
Multimodal Functions	4	10	[-100,100]	400
	5	10	[-100,100]	500
	6	10	[-100,100]	600
	7	10	[-100,100]	700
	8	10	[-100,100]	800
	9	10	[-100,100]	900
	10	10	[-100,100]	1000
Hybrid Functions	11	10	[-100,100]	1100
	12	10	[-100,100]	1200
	13	10	[-100,100]	1300
	14	10	[-100,100]	1400
	15	10	[-100,100]	1500
	16	10	[-100,100]	1600
	17	10	[-100,100]	1700
	18	10	[-100,100]	1800
	19	10	[-100,100]	1900
	20	10	[-100,100]	2000
Composition Functions	21	10	[-100,100]	2100
	22	10	[-100,100]	2200
	23	10	[-100,100]	2300
	24	10	[-100,100]	2400
	25	10	[-100,100]	2500
	26	10	[-100,100]	2600
	27	10	[-100,100]	2700
	28	10	[-100,100]	2800
	29	10	[-100,100]	2900
	30	10	[-100,100]	3000

TABLE III
PROPERTIES AND SUMMARY OF THE CEC2017 TEST FUNCTIONS

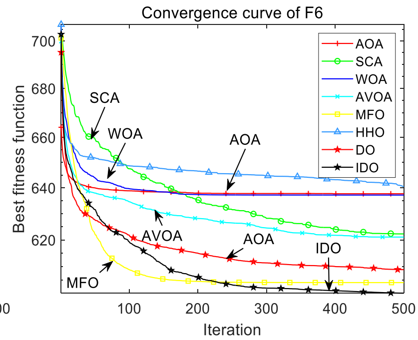
No.	Measure	AOA	SCA	WOA	AVOA	MFO	HHO	DO	IDO
1	Avg	8.9432E+09	1.1223E+09	4.3269E+08	**4.6442E+03**	2.5008E+08	1.8854E+06	5.3463E+03	1.0526E+04
	Std	3.7984E+09	4.6984E+08	4.2674E+08	4.2264E+03	5.7808E+08	3.3716E+06	**3.2322E+03**	1.7462E+04
2	Avg	2.8827E+11	5.8534E+07	9.4086E+08	4.5162E+03	2.7346E+08	2.8253E+06	**2.2625E+02**	2.6858E+02
	Std	1.1007E+12	8.8613E+07	3.0441E+09	4.8614E+03	6.1141E+08	5.8399E+06	**3.2796E+01**	9.7785E+01
3	Avg	1.3379E+04	3.1616E+03	3.9919E+03	4.0585E+02	9.1126E+03	6.3426E+02	**3.0052E+02**	3.0285E+02
	Std	2.9215E+03	1.7435E+03	2.8439E+03	9.9496E+01	1.0931E+04	2.5531E+02	**1.2044E-01**	3.7809E+00
4	Avg	1.2336E+03	4.6514E+02	4.5031E+02	4.1705E+02	4.1817E+02	4.3767E+02	**4.0659E+02**	4.1048E+02
	Std	6.2739E+02	2.9947E+01	4.3959E+01	2.7659E+01	2.6268E+01	4.5360E+01	**2.1554E+00**	1.5431E+01
5	Avg	5.6428E+02	5.5248E+02	5.3259E+02	5.4566E+02	5.3042E+02	5.6326E+02	5.3547E+02	**5.2097E+02**
	Std	2.0742E+01	1.0724E+01	1.2770E+01	1.7098E+01	1.3425E+01	2.0018E+01	1.1810E+01	**7.9248E+00**
6	Avg	6.3858E+02	6.2231E+02	6.1011E+02	6.2131E+02	6.0490E+02	6.4189E+02	6.0946E+02	**6.0064E+02**
	Std	7.0163E+00	4.2480E+00	5.8867E+00	1.3554E+01	3.7147E+00	1.1176E+01	7.7789E+00	**3.5648E-01**
7	Avg	8.0042E+02	7.8557E+02	7.6079E+02	7.6680E+02	7.4479E+02	7.9589E+02	7.5237E+02	**7.3447E+02**
	Std	1.5614E+01	1.0215E+01	1.5657E+01	1.9957E+01	1.4689E+01	2.2768E+01	1.5068E+01	**8.8527E+00**
8	Avg	8.3915E+02	8.4615E+02	8.2993E+02	8.2969E+02	8.3647E+02	8.3414E+02	8.2414E+02	**8.1824E+02**
	Std	9.1268E+00	9.1689E+00	1.1425E+01	9.1526E+00	1.3768E+01	7.8663E+00	9.6184E+00	**6.2745E+00**
9	Avg	1.4067E+03	1.1122E+03	9.9627E+02	1.2852E+03	1.0478E+03	1.6026E+03	9.9832E+02	**9.0284E+02**
	Std	2.2426E+02	1.3963E+02	9.3767E+01	2.7963E+02	2.8678E+02	1.9473E+02	1.5237E+02	6.6373E+00
10	Avg	2.3646E+03	2.4796E+03	2.0367E+03	1.9978E+03	1.9807E+03	2.0685E+03	1.8004E+03	**1.5867E+03**
	Std	2.3267E+02	**1.8468E+02**	3.8557E+02	3.4396E+02	2.8397E+02	2.8850E+02	3.5285E+02	2.4953E+02
11	Avg	3.2625E+03	1.2780E+03	1.1878E+03	1.1996E+03	1.3978E+03	1.2007E+03	1.1374E+03	**1.1296E+03**
	Std	2.0325E+03	7.0709E+01	7.1989E+01	8.3446E+01	6.0457E+02	7.8496E+01	**1.8726E+01**	4.0554E+01
12	Avg	2.8190E+08	3.0497E+07	1.2846E+06	1.8346E+06	1.6675E+06	4.6653E+06	**4.4656E+05**	5.5507E+05
	Std	3.5035E+08	2.1970E+07	1.8389E+06	1.8490E+06	3.0362E+06	4.8532E+06	**5.9352E+05**	6.1707E+05
13	Avg	1.2553E+04	7.1207E+04	**1.0632E+04**	1.3697E+04	1.1251E+04	1.7026E+04	1.1352E+04	1.1267E+04
	Std	**7.0435E+03**	6.5374E+04	7.6757E+03	8.5359E+03	1.1521E+04	1.2237E+04	7.8017E+03	1.0835E+04
14	Avg	6.7353E+03	2.2896E+03	3.4885E+03	2.8249E+03	4.9132E+03	2.3984E+03	2.9917E+03	**1.6626E+03**
	Std	5.8625E+03	1.1064E+03	1.7734E+03	1.4770E+03	4.8893E+03	1.1748E+03	1.8921E+03	**3.9933E+02**
15	Avg	1.7725E+04	4.2236E+03	4.8834E+03	5.8525E+03	7.9359E+03	8.5884E+03	3.8895E+03	**3.3132E+03**
	Std	5.8166E+03	**1.6922E+03**	2.4556E+03	3.9679E+03	7.3779E+03	3.4694E+03	2.7922E+03	2.3625E+03
16	Avg	2.1222E+03	1.8126E+03	1.8089E+03	1.8639E+03	1.7418E+03	1.9248E+03	1.7742E+03	**1.7274E+03**
	Std	1.2568E+02	**1.0569E+02**	1.5345E+02	1.8159E+02	1.1515E+02	1.1694E+02	1.2253E+02	1.3937E+02
17	Avg	1.8877E+03	1.8090E+03	1.7867E+03	1.7822E+03	1.7828E+03	1.7894E+03	1.7885E+03	**1.7597E+03**
	Std	1.1185E+02	2.1939E+01	4.1847E+01	4.0268E+01	5.0882E+01	3.9493E+01	4.6363E+01	**2.1685E+01**
18	Avg	1.7667E+07	3.4959E+05	3.8845E+04	1.6958E+04	1.9827E+04	1.6821E+04	1.7254E+04	2.7758E+04
	Std	8.9714E+07	2.7490E+05	**9.2412E+03**	1.2380E+04	1.2352E+04	1.1727E+04	1.2697E+04	1.4206E+04
19	Avg	2.1409E+05	1.1378E+04	2.9814E+04	8.9284E+03	1.5259E+04	1.3736E+04	9.1525E+03	**6.5964E+03**
	Std	4.8296E+05	9.3544E+03	7.0856E+04	6.7025E+03	1.2959E+04	1.2214E+04	7.3442E+03	**6.2236E+03**
20	Avg	2.1706E+03	2.1247E+03	2.1372E+03	2.1417E+03	2.0930E+03	2.1873E+03	2.1065E+03	**2.0344E+03**
	Std	7.9360E+01	2.9745E+01	7.1921E+01	7.3418E+01	5.8348E+01	8.1137E+01	6.0793E+01	**1.6345E+01**
21	Avg	2.3464E+03	2.2913E+03	2.3306E+03	2.3072E+03	2.3084E+03	2.3206E+03	2.3155E+03	**2.2832E+03**
	Std	4.0807E+01	6.9613E+01	**2.3761E+01**	6.9327E+01	5.8083E+01	6.7206E+01	4.6613E+01	5.8788E+01
22	Avg	3.0840E+03	2.4166E+03	2.4252E+03	**2.3028E+03**	2.3137E+03	2.3265E+03	2.4125E+03	2.3253E+03
	Std	3.8058E+02	6.6925E+01	1.9685E+02	2.1426E+01	1.9384E+01	**4.4853E+00**	3.4244E+02	1.0408E+02
23	Avg	2.7553E+03	2.6627E+03	2.6484E+03	2.6478E+03	2.6328E+03	2.6931E+03	2.6544E+03	**2.6309E+03**
	Std	4.0325E+01	1.0713E+01	1.2486E+01	2.2937E+01	1.3117E+01	2.7214E+01	1.6021E+01	**8.1805E+00**
24	Avg	2.8627E+03	2.7912E+03	2.7780E+03	**2.7478E+03**	2.7626E+03	2.8285E+03	2.7663E+03	2.7607E+03
	Std	6.9528E+01	**9.8514E+00**	2.5378E+01	8.5057E+01	4.9594E+01	1.2173E+02	7.4653E+01	1.3780E+01
25	Avg	3.4428E+03	2.9977E+03	2.9544E+03	2.9357E+03	2.9339E+03	2.9284E+03	**2.9297E+03**	2.9364E+03
	Std	2.1128E+02	2.7556E+01	3.6289E+01	3.5927E+01	2.7136E+01	6.1342E+01	6.4009E+01	**2.3478E+01**
26	Avg	4.0495E+03	3.1242E+03	3.1449E+03	3.3225E+03	3.0795E+03	3.6619E+03	3.1486E+03	**2.9745E+03**
	Std	2.9740E+02	**4.8242E+01**	2.2656E+02	5.6945E+02	1.7660E+02	5.3595E+02	4.6086E+02	1.9618E+02
27	Avg	3.2589E+03	3.1123E+03	3.1089E+03	3.1183E+03	**3.1006E+03**	3.1753E+03	3.1163E+03	3.1022E+03
	Std	5.3429E+01	3.6345E+00	1.3263E+01	2.5930E+01	**3.6037E+00**	4.9532E+01	2.1369E+01	6.7095E+00
28	Avg	3.7949E+03	3.3356E+03	3.4079E+03	3.3285E+03	3.3768E+03	3.4629E+03	3.3196E+03	**3.2847E+03**
	Std	1.9337E+02	8.7167E+01	1.2524E+02	1.4825E+02	**7.9288E+01**	1.6094E+02	1.2996E+02	1.1558E+02
29	Avg	3.4569E+03	3.2667E+03	3.2467E+03	3.3063E+03	3.2337E+03	3.3542E+03	3.2595E+03	**3.2190E+03**
	Std	1.4490E+02	**4.9978E+01**	6.0389E+01	7.4757E+01	6.0358E+01	1.2282E+02	6.2194E+01	6.0080E+01
30	Avg	3.0483E+07	1.8690E+06	2.2106E+06	4.7748E+05	7.8726E+05	3.3039E+06	5.4836E+05	**3.1783E+05**
	Std	3.2330E+07	9.4689E+05	2.5047E+06	6.2497E+05	6.8532E+05	3.5086E+06	5.8356E+05	**4.9128E+05**
Friedman Rank		7.6000	5.2667	6.0333	3.5000	3.6667	5.4333	2.8000	**1.7000**
Rank		8	5	7	3	4	6	2	**1**



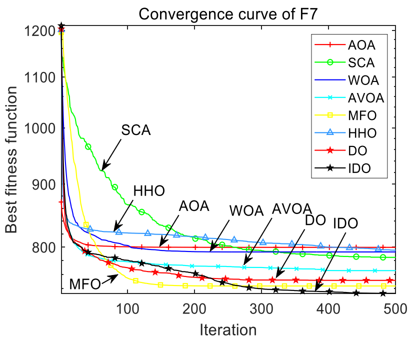
(a) F3



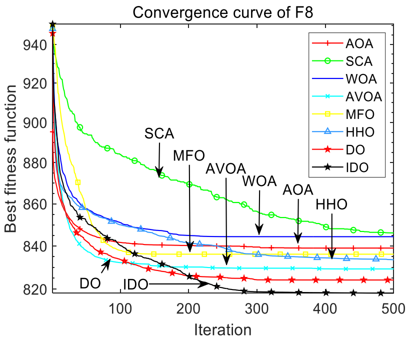
(b) F5



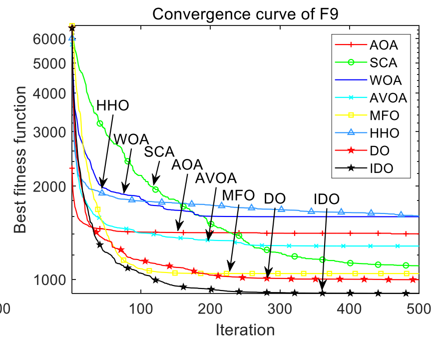
(c) F6



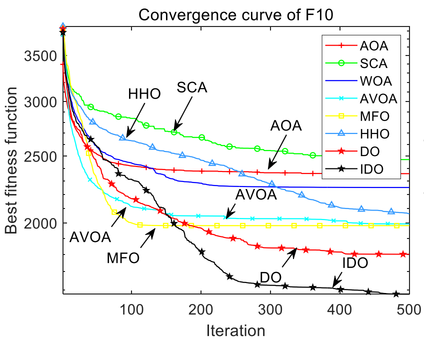
(d) F7



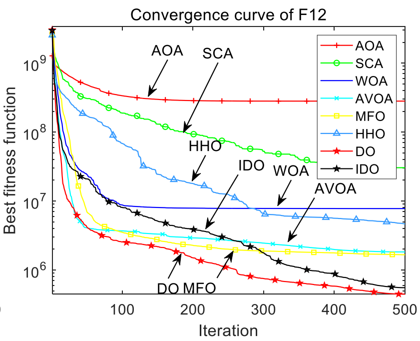
(e) F8



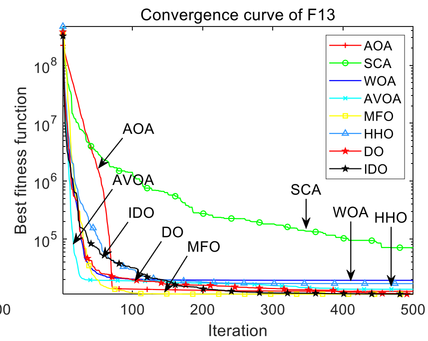
(f) F9



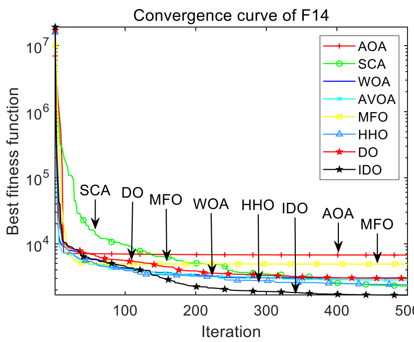
(g) F10



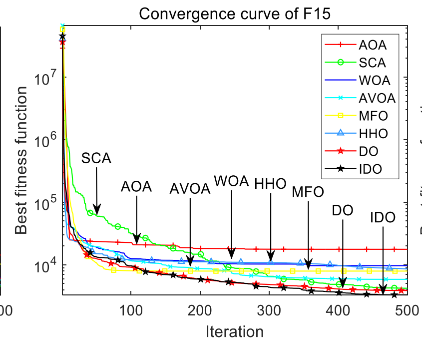
(h) F12



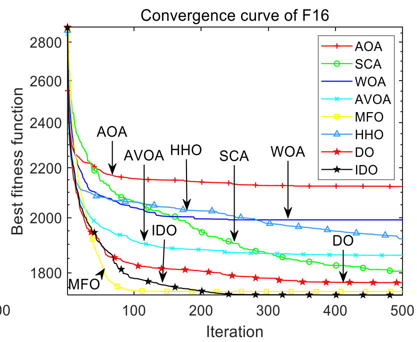
(i) F13



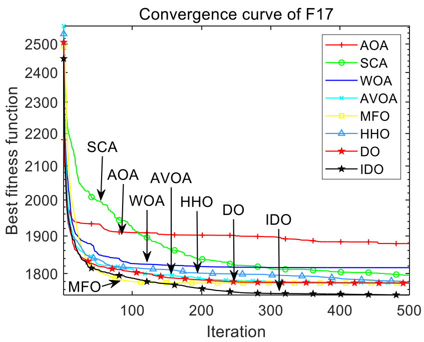
(j) F14



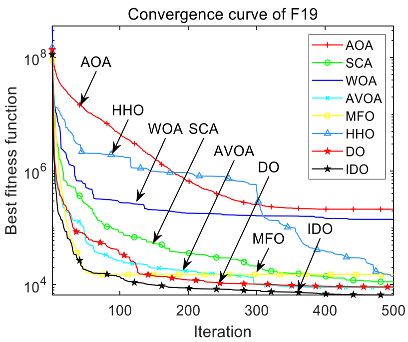
(k) F15



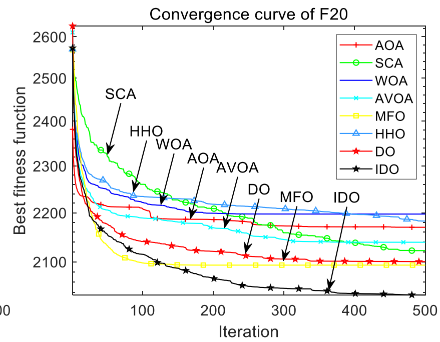
(l) F16



(m) F17



(n) F19



(o) F20

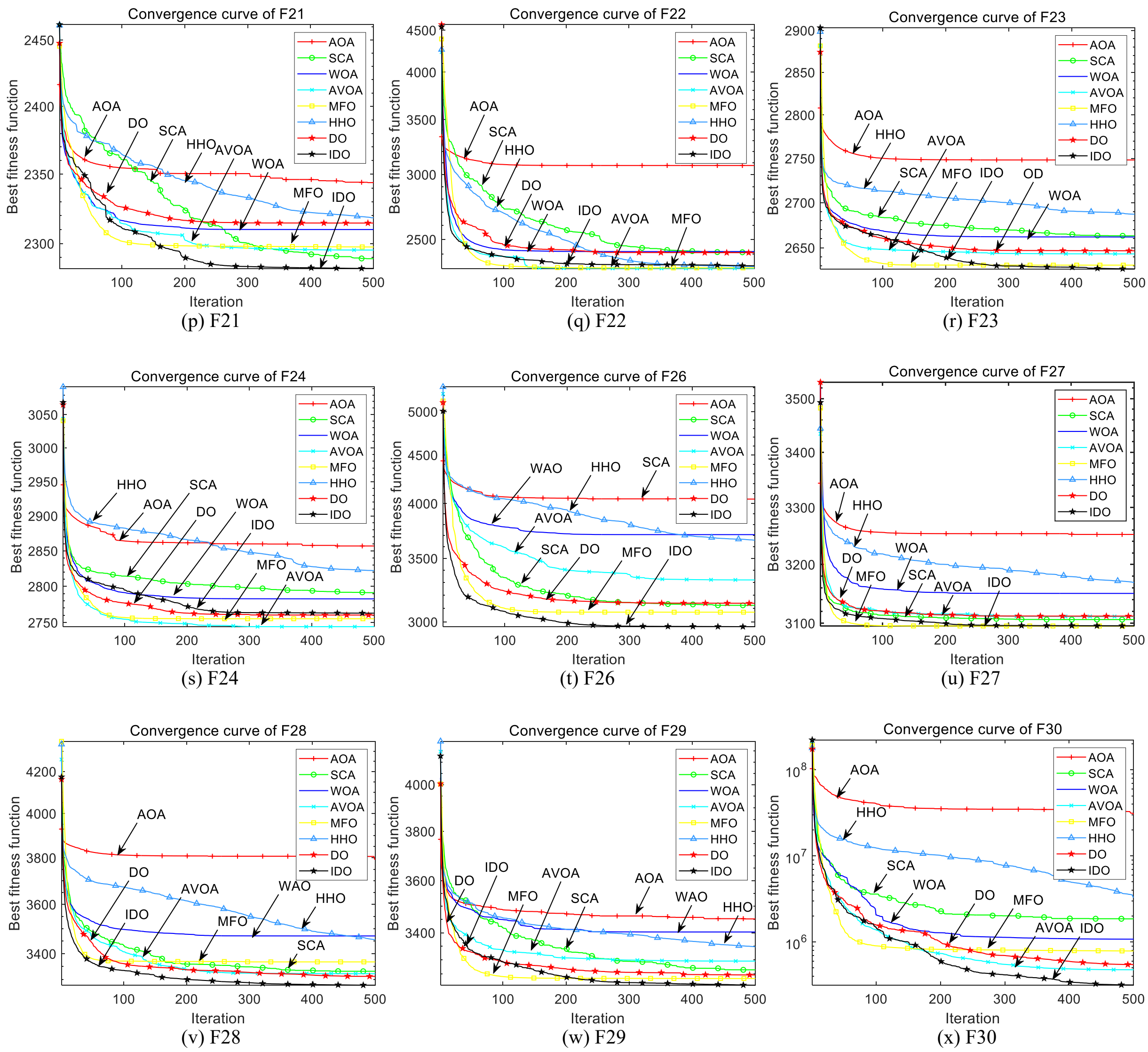


Fig. 3. Comparison of convergence curves for CEC-2017 benchmark functions

C. The performance of the IDO on the CEC2020

Description of CEC2020

To further investigate the ability of the algorithm IDO to handle complex problems, it was validated using the competitive problem CEC2020. The result will reflect the performance of the IDO algorithm in optimizing complex problems. Table IV provides a specific description of CEC2020, with different parameter settings as above.

Comparison with other optimization algorithms on CEC2020

Table V shows the average and variance of 30 runs of 30 test functions, and the optimal results of the algorithm are shown in bold. Fig. 4 shows the convergence diagram of representative functions. In this section of the experiment, the proposed algorithm IDO performed best on the vast majority of the test sets. However, there are still some algorithms, such as the SCA algorithm and the original DO algorithm, that can achieve optimal values. The experimental results are sufficient to demonstrate the superior performance of the IDO algorithm in dealing with complex problems.

Table V shows a complete comparison of the average Friedman results of all algorithms for the CEC2020 test function. From the graph, it can be seen that the algorithm proposed in this paper is among the top eight function groups in CEC2020. From the diagram, it can be seen that the algorithm proposed in this article can solve complex problems and replace previous algorithms.

TABLE IV
PROPERTIES AND SUMMARY OF THE CEC2020 TEST FUNCTIONS

Func.	No.	Dim	Range	f_{min}
Unimodal Functions	1	10	[-100,100]	100
Basic Functions	2	10	[-100,100]	1100
	3	10	[-100,100]	700
	4	10	[-100,100]	1900
Hybrid Functions	5	10	[-100,100]	1700
	6	10	[-100,100]	1600
	7	10	[-100,100]	2100
Composition Functions	8	10	[-100,100]	2200
	9	10	[-100,100]	2400
	10	10	[-100,100]	2500

TABLE V
PROPERTIES AND SUMMARY OF THE CEC2020 TEST FUNCTIONS

No.	Measure	AOA	SCA	WOA	AVOA	MFO	HHO	DO	IDO
CEC-01	Avg	9.1744E+09	1.1569E+09	7.7475E+07	**2.8688E+03**	1.9385E+08	1.0503E+07	4.1156E+03	7.3458E+03
	Std	4.4878E+09	3.9564E+08	1.1203E+08	**3.1167E+03**	4.7475E+08	3.4062E+07	3.2853E+03	4.5874E+03
CEC-02	Avg	2.3113E+03	2.4667E+03	2.2031E+03	1.9789E+03	2.0097E+03	2.1327E+03	1.8543E+03	**1.6093E+03**
	Std	2.9090+02	2.4974E+02	2.8115E+02	3.1352E+02	2.9963E+02	3.0827E+02	2.9156E+02	**2.1867E+02**
CEC-03	Avg	8.03534+02	7.8374E+02	7.8156E+02	7.7174E+02	7.3858E+02	7.8427E+02	7.5623E+02	**7.3383E+02**
	Std	1.4793E+01	1.0878E+01	1.7531E+01	1.9144E+01	1.1808E+01	2.1548E+01	1.5032E+01	**9.7926E+00**
CEC-04	Avg	2.1137E+05	1.9558E+03	1.9115E+03	1.9007E+03	1.9096E+03	1.9148E+03	1.9043E+03	**1.9052E+03**
	Std	1.7426E+05	3.9295E+01	6.9966E+00	2.0257E+00	4.9764E+00	3.1052E+00	1.5365E+00	**6.5315E-01**
CEC-05	Avg	4.5709E+05	7.7838E+04	4.4509E+05	4.2385E+04	1.8842E+05	7.1873E+04	**1.9086E+04**	1.9569E+04
	Std	1.7246E+05	1.0108E+05	7.3963E+05	5.6549E+04	2.4458E+05	6.9573E+04	**3.6789E+04**	5.3196E+04
CEC-06	Avg	1.6228E+03	1.6053E+03	1.6142E+03	1.6046E+03	1.6028E+03	1.6231E+03	1.6008E+03	**1.6062E+03**
	Std	9.2296E+00	6.9053E-01	1.3728E+01	6.4542E+00	4.2328E+00	1.4117E+01	4.2374E+00	**2.9728E-01**
CEC-07	Avg	5.3304E+05	1.5067E+04	3.3028E+05	1.1228E+04	3.8136E+04	7.5017E+04	9.1526E+03	**7.5373E+03**
	Std	1.3879E+06	8.2489E+03	5.2726E+05	8.4858E+03	7.1528E+04	1.8183E+05	6.6047E+03	**5.3573E+03**
CEC-08	Avg	3.1037E+03	2.4128E+03	2.4327E+03	2.3463E+03	2.3185E+03	2.3884E+03	2.3556E+03	**2.3006E+03**
	Std	3.0973E+02	5.2537E+01	3.9644E+02	1.5632E+02	2.2325E+01	2.5508E+02	1.8779E+02	**1.1453E+00**
CEC-09	Avg	2.8532E+03	2.7927E+03	2.7990E+03	2.7621E+03	2.7628E+03	2.8084E+03	2.7762E+03	**2.7338E+03**
	Std	5.1762E+01	**1.0927E+01**	4.9759E+01	7.1252E+01	3.6148E+01	1.2873E+02	5.5358E+01	7.9427E+01
CEC-10	Avg	3.3884E+03	2.9809E+03	2.9680E+03	2.9464E+03	2.9579E+03	2.9473E+03	2.9335E+03	**2.9163E+03**
	Std	2.0274E+02	2.3937E+01	3.5200E+01	2.6666E+01	3.6430E+01	2.8524E+01	**2.3375E+01**	6.3497E+01
Friedman Rank		7.8000	5.8000	6.0000	3.3000	3.9000	5.4000	2.5000	1.3000
Rank		8	7	4	5	6	2	3	1

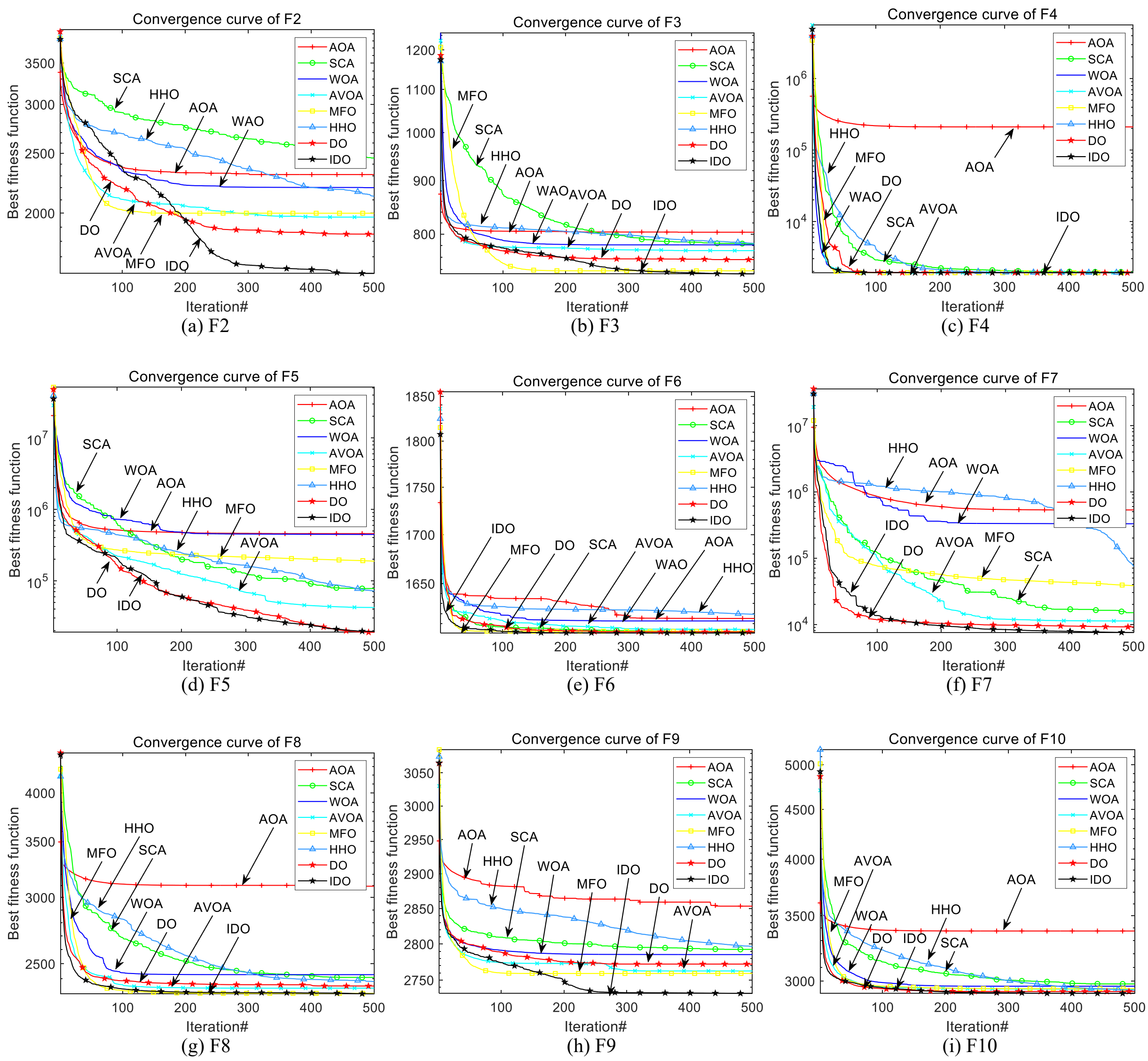


Fig. 4. Comparison of convergence curves for CEC-2020 benchmark functions

D. The performance of the IDO on the CEC2022

Description of CEC2022

CEC2022 is currently a relatively new test set whose functional characteristics are listed in Table VI. It can be used to effectively evaluate the performance of algorithms. To comprehensively evaluate the ability of IDO to explore, exploit, and avoid local minima, these features were tested and compared with other known algorithms.

Comparison with other optimization algorithms on CEC2022

IDO was tested in the CEC2022 competition, as shown in Table VII. Fig. 5 shows the convergence curves of all algorithms. From the data in Table VII, the experimental results show that IDO outperforms the other algorithms and achieves the optimal values on half of the dataset, including F3, F4, F5, F7, F8, and F11. Overall, the performance of IDO is significantly better than that of DO, except for its unsatisfactory performance in F1, F8, and F9, which is better than DO in all other test functions. Table VII presents the results of the Friedman test, which shows that the IDO algorithm ranks first with absolute advantage, the MFO ranks second, and the DO algorithm ranks third, fully demonstrating the effectiveness of the improvement strategy.

TABLE VI
PROPERTIES AND SUMMARY OF THE CEC2022 TEST FUNCTIONS

Func	No.	Dim	Range	f_{min}
Unimodal Functions	1	10	[-100,100]	300
Basic Functions	2	10	[-100,100]	400
	3	10	[-100,100]	600
	4	10	[-100,100]	800
	5	10	[-100,100]	900
Hybrid Functions	6	10	[-100,100]	1800
	7	10	[-100,100]	2000
	8	10	[-100,100]	2200
Composition Functions	9	10	[-100,100]	2300
	10	10	[-100,100]	2400
	11	10	[-100,100]	5260
	12	10	[-100,100]	2700

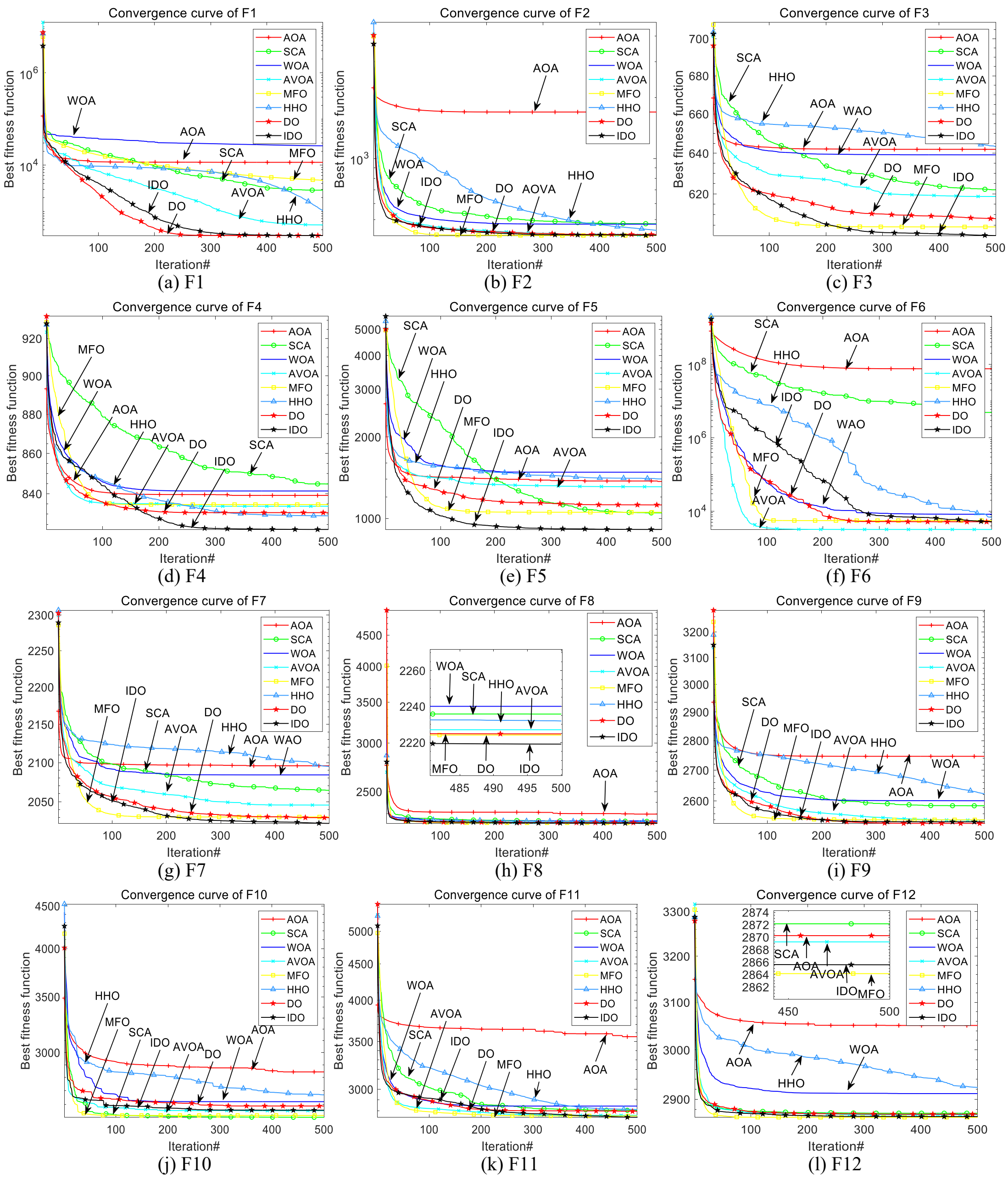


Fig. 5. Comparison of convergence curves for CEC-2022 benchmark functions

TABLE VII
PROPERTIES AND SUMMARY OF THE CEC2022 TEST FUNCTIONS

No.	Measure	AOA	SCA	WOA	AVOA	MFO	HHO	DO	IDO
CEC-01	Avg	1.1666E+04	2.8805E+03	2.6468E+04	5.1124E+02	4.7086E+03	1.0436E+03	**3.0074E+02**	3.0003E+02
	Std	4.8245E+03	1.2567E+03	1.3309E+04	2.5123E+02	6.3897E+03	3.0994E+02	**1.5845E-01**	6.2609E-01
CEC-02	Avg	1.6960E+03	4.7727E+02	4.7563E+02	4.1952E+02	**4.1772E+02**	4.4353E+02	4.2314E+02	4.1970E+02
	Std	6.4312E+02	**2.0538E+01**	9.3748E+01	2.6364E+01	2.2326E+01	3.8653E+01	3.0213E+01	2.7205E+01
CEC-03	Avg	6.4216E+02	6.2208E+02	6.3957E+02	6.1953E+02	6.0468E+02	6.4337E+02	6.0862E+02	**6.0040E+02**
	Std	7.9769E+00	4.0563E+00	1.5716E+01	1.1134E+01	5.2327E+00	1.1783E+01	7.22E+00	**1.4582E-01**
CEC-04	Avg	8.3925E+02	8.4552E+02	8.4184E+02	8.3474E+02	8.3506E+02	8.2927E+02	8.3188E+02	**8.2327E+02**
	Std	9.2726E+00	**5.9782E+00**	1.4273E+01	9.8937E+00	1.4056E+01	1.0874E+01	1.213E+01	8.3209E+00
CEC-05	Avg	1.3789E+03	1.0595E+03	1.4814E+03	1.31E+03	1.0579E+03	1.4096E+03	1.1246E+03	**9.1073E+02**
	Std	1.7056E+02	6.2806E+01	3.3357E+02	1.9577E+02	2.2304E+02	2.1852E+02	2.8574E+02	**2.5533E+01**
CEC-06	Avg	7.5699E+07	4.8862E+06	8.2367E+03	**3.1794E+03**	5.5164E+03	6.7628E+03	5.1356E+03	5.1948E+03
	Std	2.2529E+08	3.9007E+06	1.0208E+04	**1.2926E+03**	2.3235E+03	3.2648E+03	1.5628E+03	2.5449E+03
CEC-07	Avg	2.1039E+03	2.0663E+03	2.0835E+03	2.0562E+03	2.0325E+03	2.0947E+03	2.0353E+03	**2.0239E+03**
	Std	2.4980E+01	1.0701E+01	3.5669E+01	2.2593E+01	1.4349E+01	3.2449E+01	1.3063E+01	**2.2720E+00**
CEC-08	Avg	2.3048E+03	2.2441E+03	2.2424E+03	2.2324E+03	2.2276E+03	2.2369E+03	2.2225E+03	**2.2232E+03**
	Std	7.5825E+01	3.4041E+00	1.0757E+01	4.5567E+00	6.2409E+00	9.2623E+00	**2.2947E+00**	7.5138E+00
CEC-09	Avg	2.7547E+03	2.5864E+03	2.6009E+03	2.5442E+03	2.5475E+03	2.6238E+03	**2.5332E+03**	2.5396E+03
	Std	4.1643E+01	2.3946E+01	5.0634E+01	2.8004E+01	2.8924E+01	3.8436E+01	**2.0653E-03**	2.6869E+01
CEC-10	Avg	2.8445E+03	**2.5125E+03**	2.6223E+03	2.5507E+03	2.5223E+03	2.6736E+03	2.5979E+03	2.5653E+03
	Std	2.4298E+02	**2.6660E+01**	2.8712E+02	6.6285E+01	4.4535E+01	1.6559E+02	9.5325E+01	6.5236E+01
CEC-11	Avg	3.5689E+03	2.8026E+03	2.8414E+03	2.7618E+03	2.7576E+03	2.8146E+03	2.7983E+03	**2.7445E+03**
	Std	3.6224E+02	**8.5625E+01**	1.5868E+02	1.7124E+02	1.1612E+02	1.6144E+02	2.7443E+02	1.9983E+02
CEC-12	Avg	3.0505E+03	2.8716E+03	2.9124E+03	2.8774E+03	**2.8616E+03**	2.9256E+03	2.8775E+03	2.8795E+03
	Std	7.8512E+01	**1.9708E+00**	4.3064E+01	7.6765E+00	1.9956E+00	4.1732E+01	9.4188E+00	4.8566E+00
Friedman Rank		7.5000	5.0833	6.5833	3.2500	2.9167	5.8333	3.0000	1.8333
Rank		8	5	7	4	2	6	3	1

E. The performance of the proposed algorithm on real problems

In this section, IDO is used to solve three constrained engineering optimization problems to reflect its capacity to solve constrained engineering optimization problems, and the IDO algorithm is compared with other algorithms. The parameter settings of the algorithm are detailed in Table I, and the number of iterations is 500.

Three-bar Truss Design Problem

The three-bar truss design problem is particularly important mechanical constraint problems, aiming to achieve the minimum weight under the constraint conditions. The optimization process is mainly carried out for the two parameters (X_1 and X_2) of the cross-sectional area. The model illustrating the studied problem is shown in Fig. 6. The three-bar truss design problem is constrained by three inequalities, and the constraints are as follows.

Objective function: $f(X) = (2\sqrt{2}X_1 + X_2)\ l$

Constraints : $g_1(X) = \dfrac{\sqrt{2}X_1 + X_2}{\sqrt{2}X_1^2 + 2X_1X_2}P - \rho \le 0$

$$g_2(X) = \frac{X_2}{\sqrt{2}X_1^2 + 2X_1X_2}P - \rho \le 0$$

$$g_3(X) = \frac{1}{\sqrt{2}X_2 + X_1}P - \rho \le 0$$

where, variable X_1 and X_2 is the cross-sectional area of rod 1 and 2, $0 \le X_1, X_2 \le 1$, l =100cm, P =2KN/cm, σ=2KN/cm.

The convergence curve of IDO and other swarm intelligent optimization algorithms for the design problem of a three-bar truss is shown in in Fig. 7. Each algorithm independently runs this problem 30 times and obtains the optimal values, mean, and standard deviation, as shown in Table VIII. The optimum experimental data are also shown in bold in the table. The experiment showed that IDO ranked first in 30 runs, while the original DO ranks second. The algorithms IDO, DO, WOA, AVOA, MFO and HHO all found the same minimum value, with the minimum standard deviation coming from IDO, which demonstrates the robustness of IDO algorithm.

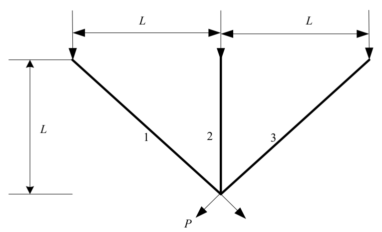


Fig. 6. Three-bar truss design problem model

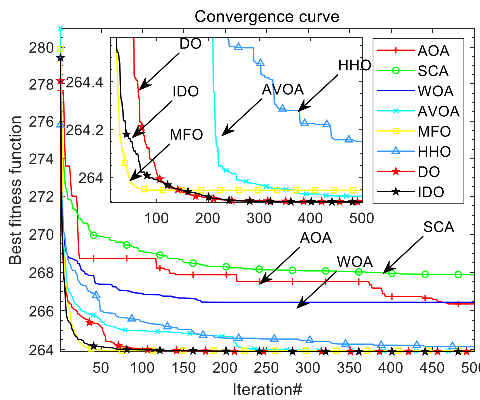


Fig.7. Convergence diagram of three-bar truss optimized by IDO and other algorithms.

Welded Beam Problem

This problem is concerned with the minimization of welded beams under certain constraints and the reduction of manufacturing costs by optimizing process parameters. The constraint conditions for this problem include bending stress, buckling load, shear stress, etc. The model diagram of this optimization problem is shown in Fig. 8, and the mathematical expressions for its objective function and constraint conditions are as follows.

Objective function:

$$f(X)=1.1047X_1^2X_2+0.04811X_3X_4(14.0+X_2)$$

Constraints:
$$g_1(X)=\tau(X)-\tau_{\max}\le 0$$
$$g_2(X)=\sigma(X)-\sigma_{\max}\le 0$$
$$g_3(X)=\delta(X)-\delta_{\max}\le 0$$
$$g_4(X)=X_1-X_4\le 0$$
$$g_5(X)=P-P_c(X)\le 0$$
$$g_6(X)=0.125-X_1\le 0$$
$$g_7(X)=1.10471X_1^2+0.04811X_3X_4(14.0+X_2)-5.0\le 0$$

Where,

$$\tau(X)=\sqrt{(\tau'^2+2\tau'\tau''X_2/(2R)+\tau''^2)},\ \tau'=P/\sqrt{2X_1X_2},$$

$$\tau''=MR/J,\ M=P(L+\frac{X_2}{2}),\quad R=\sqrt{(\frac{X_2^2}{4}+((\frac{X_1+X_3}{2})^2)},$$

$$J=2\sqrt{2X_1X_2}\left[X_2^2/4+((X_1+X_3)/2)^2\right],$$

$$\sigma(X)=6PL/X_3^2X_4,\ \delta=6PL^3/EX_3^2X_4,$$

$$P_c=4.013E\sqrt{\left(X_3^2X_4^6/36\right)}/L^2\left(1-X_3/2L\sqrt{E/4G}\right)$$

$P=6000$ lb, $L=14$ in, $E=30\times10^6$ psi,

$G=12\times10^6$ psi, $\tau_{\max}=13600$ psi, $\sigma_{\max}=30000$ psi; X_1 is the thickness of the weld, ranging from $0.1\le X_2\le 2$; X_2 is the length of the attachment part of the steel bar, ranging from $0.1\le X_2\le 10$; X_3 is the height, ranging from $0.1\le X_3\le 10$; X_4 is the thickness, ranging from $0.1\le X_4\le 2$.

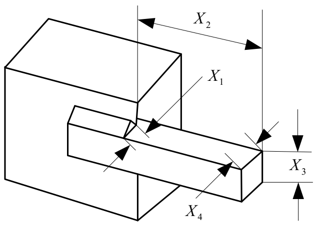


Fig. 8. Welded beam design problem model

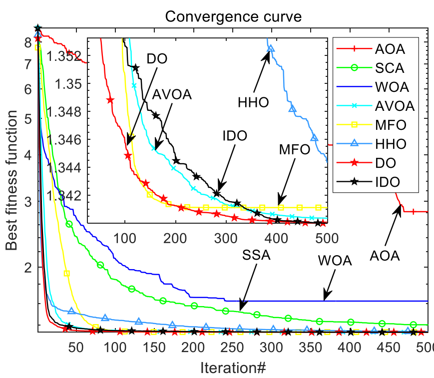


Fig. 9. Convergence diagram of the welded design problem optimized by IDO and other algorithms.

The experimental results of the IDO algorithm and other contrast algorithms on the design welded beam are shown in Fig. 9. Each algorithm independently runs this problem 30 times, obtaining the optimal values, mean, and standard deviation as shown in Table IX. The best experimental data obtained in the table is also bolded. It is evident that IDO performs best in terms of mean and standard deviation, with DO reaching the minimum value.

Pressure Vessel Problem

The goal of this problem is to optimize the production costs of containers under given constraints, such as forming costs, consumables costs, and welding costs. The model diagram of the problem is shown in Fig. 10. The mathematical function expression and constraint conditions for this problem are as follows.

Objective function:

$$f(X)=0.6224X_1X_3X_4+1.7781X_2X_3^2+3.1661X_1^2X_4+19.84X_1^2X_3$$

Constraints:
$$g_1(X)=0.0193X_3-X_1\le 0$$
$$g_2(X)=0.00954X_3-X_2\le 0$$
$$g_3(X)=1296000-\pi X_3^2X_4-\frac{4}{3}\pi X_3^3\le 0$$
$$g_4(X)=X_4-240\le 0$$

where, X_1 represents the cylinder head (Th) and X_2 represents the cylinder wall thickness (Ts). Among them $0.0625\le X_1$, $X_2\le 6.1875$, X_3 represents the radius (R) of the cylinder and cylinder head, X_4 represents the length of the cylinder (L). Among these four variables, X_1 and X_2 is a uniformly discrete variable with an interval of 0.0625, X_3 and X_4 is a continuous variable.

The experimental results of IDO and other algorithms on pressure vessel design problems are shown in Fig.11. Each algorithm independently runs this problem 30 times, obtaining the best values, average value, and standard deviation as shown in Table X. The best experimental data obtained in the table are also bolded. The experiment showed that IDO achieved the minimum value in a single experiment. The IDO ranked first in terms of average and variance in 30 runs, demonstrating the robustness of this method in solving practical problems.

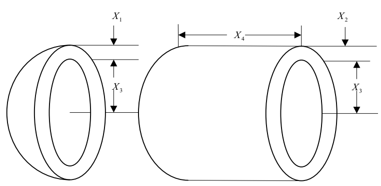


Fig. 10. Pressure vessel design problem model

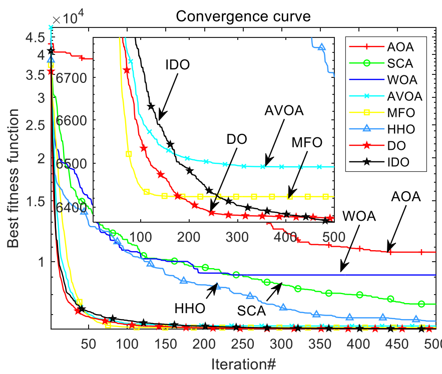


Fig. 11. Convergence diagram of the pressure vessel design problem optimized by IDO and other algorithms.

TABLE VIII
SOLUTION OF IDO AND OTHER ALGORITHMS FOR THE THREE-BAR TRUSS DESIGN PROBLEM

	AOA	SCA	WOA	AVOA	MFO	HHO	DO	IDO
$f(x)$	264.033	263.914	263.896	263.896	263.896	263.896	263.896	263.896
X_1	0.7928	0.7902	0.7874	0.7882	0.7885	0.7890	0.7887	0.7887
X_2	0.3975	0.4039	0.4119	0.4096	0.4087	0.4074	0.4082	0.4083
Best	264.033	263.914	**263.896**	**263.896**	**263.896**	**263.896**	**263.896**	**263.896**
Ave	266.355	267.872	266.442	263.921	263.946	264.150	263.898	**263.897**
Std	4.6561	7.6153	3.9729	0.0411	0.1106	0.3180	0.0031	**0.0020**

TABLE IX
SOLUTION OF IDO AND OTHER ALGORITHMS FOR THE WELDED BEAM PROBLEM

	AOA	SCA	WOA	AVOA	MFO	HHO	DO	IDO
$f(x)$	1.489438	1.355169	1.359921	1.339991	1.340052	1.340753	**1.339957**	1.339966
X_1	0.1635	0.1738	0.1640	0.1826	0.1830	0.1638	0.1826	0.1830
X_2	2.7963	2.6168	2.8222	2.4145	2.4073	2.7647	2.4129	2.4075
X_3	10.0000	10.0000	9.3695	9.5759	9.5818	9.8375	9.5818	9.5819
X_4	0.1859	0.1818	0.1914	0.1832	0.1830	0.1814	0.1830	0.1830
Best	1.489438	1.355169	1.359921	1.339991	1.340052	1.340753	**1.339957**	1.339961
Ave	2.815356	1.402795	1.623356	1.340362	1.341116	1.344331	1.340004	**1.340003**
Std	1.11E+00	3.13E-02	2.53E-01	3.01E-04	1.06E-03	2.31E-03	3.45E-05	**3.36E-05**

TABLE X
SOLUTION OF IDO AND OTHER ALGORITHMS FOR THE PRESSURE VESSEL PROBLEM

	AOA	SCA	WOA	AVOA	MFO	HHO	DO	IDO
$f(x)$	6674.699	6172.894	6154.616	5882.494	5880.671	6214.174	5880.859	5880.722
X_1	0.799	0.810	0.939	0.778	0.778	0.830	0.778	0.778
X_2	0.518	0.447	0.537	0.383	0.383	0.412	0.383	0.383
X_3	40.974	40.792	48.301	40.328	40.320	42.841	40.320	40.320
X_4	200.000	200.000	112.424	199.889	200.000	167.653	199.992	200.000
Best	6674.699	6172.894	6154.616	5882.494	5880.671	6214.174	5880.859	5880.722
Ave	11741.417	7535.900	9488.520	6362.568	6414.315	6826.969	6321.864	6357.286
Std	4119.637	853.174	2749.840	379.566	490.083	371.782	497.413	536.449

V. CONCLUSIONS AND FUTURE WORKS

In this paper, we propose IDO, an improved version of the Dandelion Optimizer (DO). IDO incorporates three improvement strategies to enhance the exploration and exploitation of the algorithm. To evaluate the performance of the algorithm, we conducted experiments using three challenging standard test functions from CEC2017, CEC2020, and CEC2022. We compared the results of IDO with seven other excellent algorithms under the same experimental conditions to highlight the optimization ability of our proposed algorithm. The statistical results from the benchmark demonstrate that IDO consistently outperforms the other algorithms in terms of solution quality and convergence speed. The proposed improvement strategy allows for a more rational allocation of exploration and exploitation stages, leading to faster convergence and avoidance of local optimization. Furthermore, the results of three mechanical engineering optimization problems indicate that IDO has significant advantages in dealing with engineering optimization problems in restricted or unidentified search spaces. We comprehensively evaluate the proposed improvement strategy, the working principle, stability, and optimization ability of the algorithm. Although IDO demonstrates outstanding performance in most cases, it does have a few shortcomings. For example, when dealing with certain unimodal problems, its advantages are not significant, and there is a lack of in-depth development mechanisms. Therefore, we plan to conduct further research in these areas. In future work, we intend to apply IDO to more complex engineering optimization problems and propose both base-level and multi-objective versions of the algorithm.

REFERENCES

[1] Xie L, Han T, Zhou H, et al. Tuna swarm optimization: a novel swarm-based metaheuristic algorithm for global optimization[J]. Computational Intelligence and Neuroscience, pp.1-22, 2021.

[2] Abualigah, Laith, et al. "The arithmetic optimization algorithm." Computer Methods in Applied Mechanics and Engineering, vol.376, pp.113609, 2021.

[3] Hajipour V, Kheirkhah A S, Tavana M, et al. Novel Pareto-based meta-heuristics for solving multi-objective multi-item capacitated lot-sizing problems[J]. The International Journal of Advanced Manufacturing Technology, vol,80, pp.31-45, 2015.

[4] Dhiman G, Kumar V. Seagull optimization algorithm: Theory and its applications for large-scale industrial engineering problems[J]. Knowledge-based Systems, vol.165, pp.169-196, 2019.

[5] Karaboğa D, Ökdem S. A simple and global optimization algorithm for engineering problems: differential evolution algorithm[J]. Turkish Journal of Electrical Engineering and Computer Sciences, vol.12, no.1, pp.53-60, 2004.

[6] Leardi R, Boggia R, Terrile M. Genetic algorithms as a strategy for feature selection[J]. Journal of Chemometrics, vol.6, no.5, pp. 267-281, 1992.

[7] Kabir M M, Shahjahan M, Murase K. A new hybrid ant colony optimization algorithm for feature selection[J]. Expert Systems with Applications, vol.39, no.3, pp.3747-3763, 2012.

[8] Mafarja M M, Mirjalili S. Hybrid whale optimization algorithm with simulated annealing for feature selection[J]. Neurocomputing, vol.260, pp. 302-312, ,2017.

[9] Ragan-Kelley J, Adams A, Paris S, et al. Decoupling algorithms from schedules for easy optimization of image processing pipelines[J]. ACM Transactions on Graphics (TOG), vol.31, no.4, pp.1-12, 2012.
[10] Daly S. A visual model for optimizing the design of image processing algorithms[C]//Proceedings of 1st International Conference on Image Processing. IEEE, vol.2, pp.16-20, 1994.
[11] Alsheikh M A, Lin S, Niyato D, et al. Machine learning in wireless sensor networks: Algorithms, strategies, and applications[J]. IEEE Communications Surveys & Tutorials, vol.16, no.4, pp.1996-2018, 2014.
[12] Abbasi A A, Younis M. A survey on clustering algorithms for wireless sensor networks[J]. Computer Communications, vol.30, no.14-15, pp. 2826-2841, 2007.
[13] Yang L, Shami A. On hyperparameter optimization of machine learning algorithms: Theory and practice[J]. Neurocomputing, vol.415, pp.295-316, 2020.
[14] Daelemans W, Hoste V, De Meulder F, et al. Combined optimization of feature selection and algorithm parameters in machine learning of language[C]//Machine Learning: ECML 2003: 14th European Conference on Machine Learning, Cavtat-Dubrovnik, Croatia, September pp.22-26, 2003. Proceedings 14. Springer Berlin Heidelberg, pp.84-95, 2003.
[15] Young S R, Rose D C, Karnowski T P, et al. Optimizing deep learning hyper-parameters through an evolutionary algorithm[C]//Proceedings of the Workshop on Machine Learning in High-performance Computing Environments, pp.1-5, 2015.
[16] Xu J, Noo F. Convex optimization algorithms in medical image reconstruction—in the age of AI[J]. Physics in Medicine & Biology, vol.67, no.7, pp.07TR01, 2022.
[17] Alterovitz R, Goldberg K. Motion planning in medicine: Optimization and simulation algorithms for image-guided procedures[M]. Springer, 2008.
[18] Filip M, Zoubek T, Bumbalek R, et al. Advanced computational methods for agriculture machinery movement optimization with applications in sugarcane production[J]. Agriculture, vol.10, no.10, pp.434, 2020.
[19] Hameed I A, Bochtis D, Sørensen C A. An optimized field coverage planning approach for navigation of agricultural robots in fields involving obstacle areas[J]. International Journal of Advanced Robotic Systems, vol.10, no.5, pp.231, 2013.
[20] Ponsich A, Jaimes A L, Coello C A C. A survey on multiobjective evolutionary algorithms for the solution of the portfolio optimization problem and other finance and economics applications[J]. IEEE Transactions on Evolutionary Computation, vol.17, no.3, pp.321-344, 2012.
[21] Al Janabi M A M. Optimal and investable portfolios: An empirical analysis with scenario optimization algorithms under crisis market prospects[J]. Economic Modelling, vol.40, pp.369-381, 2014.
[22] Erol O K, Eksin I. A new optimization method: big bang–big crunch[J]. Advances in Engineering Software, vol.37, no.2, pp.106-111, 2006.
[23] Kennedy J, Eberhart R. Particle swarm optimization[C]//Proceedings of ICNN'95-International Conference on Neural Networks. IEEE, vol.4, pp.1942-1948, 1995.
[24] Mirjalili S. The ant lion optimizer[J]. Advances in Engineering Software, vol.83, pp.80-98, 2015.
[25] Mirjalili S, Mirjalili S M, Lewis A. Grey wolf optimizer[J]. Advances in Engineering Software, vol.69, pp.46-61, 2014.
[26] Faramarzi A, Heidarinejad M, Mirjalili S, et al. Marine Predators Algorithm: A nature-inspired metaheuristic[J]. Expert Systems with Applications, vol.152, pp.113377, 2020.
[27] Fonseca C M, Fleming P J. An overview of evolutionary algorithms in multiobjective optimization[J]. Evolutionary Computation, vol.3, no.1, pp. 1-16, 1995.
[28] Storn R, Price K. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces[J]. Journal of Global Optimization, vol.11, pp.341-359, 1997.
[29] Bonabeau E, Dorigo M, Theraulaz G. Swarm intelligence: from natural to artificial systems[M]. Oxford University Press, 1999.
[30] Kosorukoff A. Human based genetic algorithm[C]//2001 IEEE International Conference on Systems, Man and Cybernetics. e-systems and e-man for Cybernetics in Cyberspace (Cat. No. 01CH37236). IEEE, vol.5, pp.3464-3469, 2001.
[31] Rao R V, Savsani V J, Vakharia D P. Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems[J]. Computer-aided Design, vol.43, no.3, pp.303-315, 2011.
[32] Faramarzi A, Heidarinejad M, Stephens B, et al. Equilibrium optimizer: A novel optimization algorithm[J]. Knowledge-Based Systems, vol.191, pp.105190, 2020.
[33] Satapathy S, Naik A. Social group optimization (SGO): a new population evolutionary optimization technique[J]. Complex & Intelligent Systems, vol.2, no.3, pp.173-203, 2016.
[34] Liu S H, Mernik M, Hrnčič D, et al. A parameter control method of evolutionary algorithms using exploration and exploitation measures with a practical application for fitting Sovova's mass transfer model[J]. Applied Soft Computing, vol.13, no.9, pp.3792-3805, 2013.
[35] Shabani A, Asgarian B, Salido M, et al. Search and rescue optimization algorithm: A new optimization method for solving constrained engineering optimization problems[J]. Expert Systems with Applications, vol.161, pp.113698, 2020.
[36] Hayyolalam V, Kazem A A P. Black widow optimization algorithm: a novel meta-heuristic approach for solving engineering optimization problems[J]. Engineering Applications of Artificial Intelligence, vol.87, pp.103249, 2020.
[37] Eskandar H, Sadollah A, Bahreininejad A, et al. Water cycle algorithm–A novel metaheuristic optimization method for solving constrained engineering optimization problems[J]. Computers & Structures, vol.110, pp.151-166, 2012.
[38] Pan J S, Zhang L G, Wang R B, et al. Gannet optimization algorithm: A new metaheuristic algorithm for solving engineering optimization problems[J]. Mathematics and Computers in Simulation, vol.202, pp. 343-373, 2022.
[39] Wang L, Cao Q, Zhang Z, et al. Artificial rabbits optimization: A new bio-inspired meta-heuristic algorithm for solving engineering optimization problems[J]. Engineering Applications of Artificial Intelligence, vol.114, pp.105082, 2022.
[40] Hongyu Long，Shanghua Liu, et al，"Optimal Reactive Power Dispatch Based on Multi-strategy Improved Aquila Optimization Algorithm", IAENG International Journal of Computer Science, vol. 49, no.4, pp.1249-1267, 2022.
[41] Gonggui Chen，Feng Qin, et al，"Fuzzy PID Controller Optimized by Improved Gravitational Search Algorithm for Load Frequency Control in Multi-area Power System"．IAENG International Journal of Computer Science, vol. 49, no.1, pp125-139，2022.
[42] Zhao S, Zhang T, Ma S, et al. Dandelion Optimizer: A nature-inspired metaheuristic algorithm for engineering applications[J]. Engineering Applications of Artificial Intelligence, vol.114, pp.105075, 2022.
[43] Abbassi R, Saidi S, Abbassi A, et al. Accurate Key Parameters Estimation of PEMFCs' Models Based on Dandelion Optimization Algorithm[J]. Mathematics, vol.11, no.6, pp.1298, 2023.
[44] Kaveh A, Zaerreza A, Zaerreza J. Enhanced dandelion optimizer for optimum design of steel frames[J]. Iranian Journal of Science and Technology, Transactions of Civil Engineering, pp.1-14, 2023.
[45] Halassa E, Mazouz L, Seghiour A, et al. Revolutionizing Photovoltaic Systems: An Innovative Approach to Maximum Power Point Tracking Using Enhanced Dandelion Optimizer in Partial Shading Conditions[J]. Energies, vol.16, no.9, pp.3617, 2023.
[46] Akyol S, Yildirim M, Alatas B. CIDO: Chaotically Initialized Dandelion Optimization for Global Optimization[J]. International Journal of Advanced Networking and Applications, vol.14, no.6, pp.5696-5704, 2023.
[47] Abualigah L, Diabat A, Mirjalili S, et al. The arithmetic optimization algorithm[J]. Computer Methods in Applied Mechanics and Engineering, vol.376, pp.113609, 2021.
[48] Mirjalili S. SCA: a sine cosine algorithm for solving optimization problems[J]. Knowledge-Based Systems, vol.96, pp.120-133, 2016.
[49] Mirjalili S, Lewis A. The whale optimization algorithm[J]. Advances in Engineering Software, vol.95, pp.51-67, 2016.
[50] Abdollahzadeh B, Gharehchopogh F S, Mirjalili S. African vultures optimization algorithm: A new nature-inspired metaheuristic algorithm for global optimization problems[J]. Computers & Industrial Engineering, vol.158, pp.107408, 2021.
[51] Mirjalili S. Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm[J]. Knowledge-Based Systems, vol.89, pp. 228-249, 2015.
[52] Heidari A A, Mirjalili S, Faris H, et al. Harris hawks optimization: Algorithm and applications[J]. Future Generation Computer Systems, vol.97, pp.849-872, 2019.