Available online at www.sciencedirect.com
ScienceDirect

Mathematics and Computers in Simulation 185 (2021) 17–46

MATHEMATICS AND COMPUTERS IN SIMULATION

www.elsevier.com/locate/matcom

Original articles

Nature-inspired approach: An enhanced whale optimization algorithm for global optimization

Zheping Yan, Jinzhong Zhang*, Jia Zeng, Jialing Tang

College of Intelligent Systems Science and Engineering, Harbin Engineering University, Harbin 150001, China

Received 23 August 2020; received in revised form 26 November 2020; accepted 10 December 2020
Available online 19 December 2020

Abstract

The whale optimization algorithm is based on the bubble-net attacking behavior of humpback whales and simulates encircling prey, bubble-net attacking and searching for prey to obtain the global optimal solution. However, the basic whale optimization algorithm has the disadvantage of search stagnation, easily falls into a local optimum, has slow convergence speed and has low calculation accuracy. The Lévy flight strategy is beneficial for expanding the search range and prevents the algorithm from falling into a local optimum, which enhances the global search ability. The ranking-based mutation operator can increase the selection probability and accelerate the convergence speed to enhance the local search ability. To overcome these shortcomings and avoid premature convergence, the Lévy flight strategy and the ranking-based mutation operator are added to the whale optimization algorithm. In this paper, an enhanced whale optimization algorithm is proposed, which realizes complementary advantages to balance exploration and exploitation. Eighteen benchmark test functions and five structural engineering design problems are used to verify the robustness and overall optimization performance of the enhanced whale optimization algorithm. The experimental results show that the enhanced whale optimization algorithm is an effective and feasible method that has a fast convergence speed, high calculation accuracy, strong robustness and stability.

Keywords: Whale optimization algorithm; Lévy flight strategy; Ranking-based mutation operator; Benchmark test functions; Structural engineering design

1. Introduction

Optimization theory is an important branch in the field of mathematics, and the purpose of optimization is to obtain the global optimal solution from the numerous candidate solutions. Traditional optimization methods have certain limitations in solving complex and large-scale optimization problems, which have considerable time and optimization costs and relatively low calculation accuracy. Compared with traditional optimization methods, the meta-heuristic optimization algorithm not only has the advantages of simple implementation, strong robustness, easy expansion and self-organization but can also be effectively combined with certain unique strategies or other algorithms to balance the global search ability and the local search ability. Some meta-heuristic optimization algorithms, such as the artificial bee colony [28], the cuckoo search [18], the dragonfly algorithm [47], moth–flame

* Corresponding author.
E-mail addresses: zhangjinzhongz@126.com, zhangjinzhong@hrbeu.edu.cn (J. Zhang).

https://doi.org/10.1016/j.matcom.2020.12.008
0378-4754/

optimization [46], particle swarm optimization [10], and water wave optimization [77], have been applied in the fields of artificial intelligence, computer science and control engineering. The artificial bee colony, inspired by the behavior of honeybees in nature, is a heuristic bionic swarm intelligence algorithm. The cuckoo search simulates the brood parasitism of the cuckoo species and combines the Lévy flight strategy to effectively solve the optimization problem. The dragonfly algorithm is a novel swarm intelligence algorithm that is based on the dragonfly flight path, which avoids natural enemies and follows foraging behavior. Moth–flame optimization is a stochastic optimization algorithm that is based on the navigation mechanism of a moth to simulate the spiral flight path of a moth. Particle swarm optimization is a swarm intelligence algorithm designed by simulating bird predation behavior. Water wave optimization based on shallow water wave theory mimics propagation, refraction and breaking to find the global optimal solution.

Mirjalili et al. designed a whale optimization algorithm to solve mathematical optimization problems and structure design problems, and the results proved that the whale optimization algorithm is effective and feasible [49]. Yan et al. proposed an improved whale optimization algorithm to optimize the weights between the input layer and the competition layer of the S_Kohonen network, and the proposed algorithm had higher classification accuracy [69]. Ebrahimgol et al. introduced a whale optimization algorithm to solve a typical WWER1000 nuclear power plant problem, and the results demonstrated that the whale optimization algorithm has a good efficiency [14]. Jiang et al. applied an improved whale optimization algorithm to solve multifarious optimization problems, and the results showed that the proposed algorithm has a faster local convergence speed, higher convergence accuracy and lower computational complexity [26]. Chen et al. combined the whale optimization algorithm based on quasi-opposition with the chaos mechanism for global optimization problems, and the results indicated that the proposed algorithm had a strong global search ability to improve the convergence speed [7]. Qais et al. created an enhanced whale optimization algorithm for maximum power point tracking of variable-speed wind generators, and the results revealed that the enhanced whale optimization algorithm had strong robustness and better optimization performance [56]. Mohammadi et al. combined the whale optimization algorithm with support vector regression to solve the daily ET0 modeling problem, and the hybrid method had the best performance [53]. Cao et al. proposed an improved whale optimization algorithm to design an optimal proton exchange membrane fuel cell system, and the results showed that the proposed algorithm had higher efficiency and better stability [6]. Hou et al. proposed a multiobjective economic model predictive control method based on a quantum simultaneous whale optimization algorithm for gas turbine system control, and the results indicated that the proposed algorithm was superior in terms of desired economic performance, high precision, rapidity and robustness [23]. Sun et al. used a whale optimization algorithm based on quadratic interpolation to solve high-dimensional global optimization problems, and the results showed that the proposed algorithm had a faster convergence speed and higher solution accuracy [64]. Zhang et al. proposed a whale optimization algorithm based on community detection to discover communities in networks, and the results demonstrated that the improved whale optimization algorithm successfully detected communities and obtained more accurate results [73]. Wong et al. presented the whale optimization algorithm for optimal placement and sizing of battery energy storage systems to reduce the power losses in the distribution grid, and the whale optimization algorithm had outstanding performance [68]. Azizi et al. proposed an upgraded whale optimization algorithm for tuning the parameters of the fuzzy controller, and the proposed algorithm was capable of providing competitive results [2]. Zhang et al. applied an improved whale optimization algorithm for locating electric vehicle charging stations with service capacity, and the improved whale optimization algorithm solved practical locating planning projects and reduced social costs [75]. Luo et al. proposed a whale optimization algorithm with a ranking-based mutation operator to solve the IIR system identification problem, and the results showed that the proposed algorithm was superior to other algorithms [38]. Zhou et al. presented an improved whale optimization algorithm based on a Lévy flight strategy to solve the engineering optimization problem, and the results indicated that the proposed algorithm had better overall optimization performance [78]. Ling et al. proposed a whale optimization algorithm based on a Lévy flight strategy to solve function optimization problems, and the proposed algorithm had a faster convergence speed and higher calculation accuracy [34]. Du et al. analyzed the effects of structure and information fusion strategies, and the results demonstrated that the networked evolutionary algorithm framework significantly improved the performance [13]. Fister et al. proposed a novel search to solve global optimization, and the results showed great potential for using novel searches in global optimization [16]. Fister et al. provided a chaos-based firefly algorithm to improve the randomness and increase the diversity of the population [17]. Zhang et al. proposed a novel vector coevolving particle swarm optimization algorithm to solve benchmark functions, and

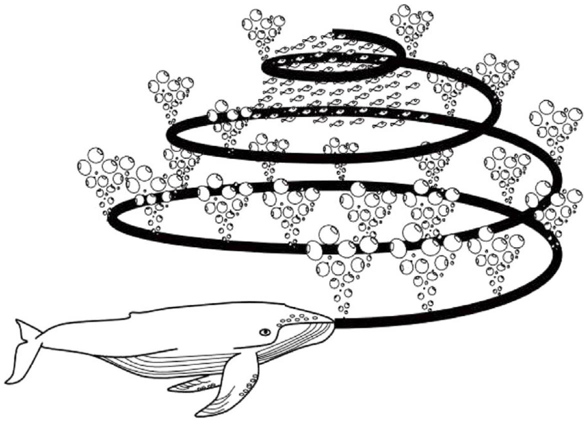


Fig. 1. Bubble-net feeding behavior of humpback whales.

the proposed algorithm had a better performance in terms of solution accuracy and statistical results [74]. Yi et al. introduced the three crossover operators of the nondominated sorting genetic algorithm, the third version, to solve large-scale optimization problems and obtain better experimental results [71].

The whale optimization algorithm, which is inspired by the bubble-net strategy, finds the global optimal solution by simulating encircling prey, bubble-net attacking and searching for prey [49]. The basic whale optimization algorithm not only easily converges prematurely and undergoes search stagnation but also has a slow convergence speed and low calculation accuracy. To improve the overall optimization ability, the Lévy flight strategy [3] and the ranking-based mutation operator [20,27] are added to the basic whale optimization algorithm. The Lévy flight strategy can expand the search space and jump out of a local optimum to improve the global search ability. The ranking-based mutation operator can accelerate the convergence speed and selection probability to improve the local search ability. The enhanced whale optimization algorithm combines the advantages of the Lévy flight strategy and the ranking-based mutation operator to effectively balance the global search ability and the local search ability and to obtain the global optimal solution. The enhanced whale optimization algorithm is applied to solve eighteen benchmark test functions and five structural engineering design problems. The experimental results show that the enhanced whale optimization algorithm not only has strong stability, robustness and global search ability but also has better optimization results compared with other algorithms.

The article is divided into the following sections. Section 2 introduces the whale optimization algorithm. Section 3 describes the enhanced whale optimization algorithm. The simulation test and result analysis are depicted in Section 4. Finally, conclusions and future research are given in Section 5.

2. Whale optimization algorithm

The whale optimization algorithm simulates the unique search method and surround hunting mechanism of humpback whales to find the optimal solution, which mainly contains three important stages: encircling prey, bubble-net attacking strategy and search for prey [49]. In the whale optimization algorithm, the position of each humpback whale represents a search agent. The whale optimization algorithm finds the optimal solution of the global optimization problem by constantly updating the search agent. The model of bubble-net feeding behavior is given in Fig. 1.

2.1. Encircling prey

Humpback whales need to determine the positions of their prey in order to surround them. Since the positions of the prey in the search space are unknown, assuming that the current optimal search agent of the whale optimization algorithm is the target prey, other humpback whales update their positions according to the position of the optimal search agent. The position update is calculated as follows:

$$\boldsymbol{D} = \left| C \cdot \boldsymbol{X}^*(t) - \boldsymbol{X}(t) \right| \tag{1}$$

$$\boldsymbol{X}(t+1) = \boldsymbol{X}^*(t) - A \cdot \boldsymbol{D} \tag{2}$$

where t is the current iteration number, $\boldsymbol{X}^*$ is the best position vector of the search agent, $\boldsymbol{X}$ is the current position vector of the search agent, $||$ represents the absolute value operation, and $\cdot$ represents elementwise multiplication. Here, A and C are two parameters and are, respectively calculated as follows:

$$A = 2a \cdot r - a \tag{3}$$

$$C = 2 \cdot r \tag{4}$$

where r is a random number in [0, 1], and a is the convergence factor, which decreases linearly from 2 to 0 as the number of iterations increases. This factor is expressed as $a = 2 - 2t/t_{\max}$, where t is the current iteration number and $t_{\max}$ is the maximum iteration number. By reducing the convergence factor a, the population is simulated to be close to the prey and enclosed by contraction.

2.2. Bubble-net attacking strategy (exploitation phase)

The bubble-net attacking strategy mainly includes two mechanisms: the shrinking encircling mechanism and logarithmic spiral position updating. The shrinking encircling mechanism updates the positions of the humpback whales in Eq. (2). The parameter A is expressed as $A = 2a \cdot r - a$. The range of fluctuation of A is affected by the convergence factor a value. The range of fluctuation of A decreases as the factor a decreases. The A is a random value in $[-a, a]$, where the factor a decreases from 2 to 0 during iteration. When iteration number t increases, the parameter A and the convergence factor a decrease, the whale hunting path shrinks and the calculation accuracy increases. If the A value is in range of $[-1, 1]$, the next position of the whale optimization algorithm is a random position between the current humpback whale and the prey. Logarithmic spiral position updating is used to calculate the distance between the humpback whale and the prey to capture the prey. The position update is calculated as follows:

$$\boldsymbol{D}' = \left|\boldsymbol{X}^*(t) - \boldsymbol{X}(t)\right| \tag{5}$$

$$\boldsymbol{X}(t+1) = \boldsymbol{D}' \cdot e^{bl} \cdot \cos(2\pi l) + \boldsymbol{X}^*(t) \tag{6}$$

where $\boldsymbol{D}'$ is the distance between the whale and its prey (the best solution obtained so far), l is a random number in $[-1, 1]$, and b is a constant that defines the shape of the logarithmic spiral.

Assume that the probability of the whale optimization algorithm choosing the shrinking encircling mechanism or logarithmic spiral position updating is 50%. The model is calculated as follows:

$$\boldsymbol{X}(t+1) = \begin{cases} \boldsymbol{X}^*(t) - A \cdot \boldsymbol{D} & if \;\; p < 0.5 \\ \boldsymbol{D}' \cdot e^{bl} \cdot \cos(2\pi l) + \boldsymbol{X}^*(t) & if \;\; p \geq 0.5 \end{cases} \tag{7}$$

where p is a random number in [0, 1].

2.3. Search for prey (exploration phase)

If $|A| > 1$, the whale optimization algorithm does not select prey to update its position but randomly searches for prey according to each other's positions. A random search agent is applied to replace the prey to perform a global search and avoid falling into a local optimum. The position is calculated as follows:

$$\boldsymbol{D} = |C \cdot \boldsymbol{X}_{rand}(t) - \boldsymbol{X}(t)| \tag{8}$$

$$\boldsymbol{X}(t+1) = \boldsymbol{X}_{rand}(t) - A \cdot \boldsymbol{D} \tag{9}$$

where $\boldsymbol{X}_{rand}$ is a random position vector (a random whale).

To better describe the solution process, the pseudocode of the whale optimization algorithm is given in Algorithm 1.

3. Enhanced whale optimization algorithm

To overcome the premature convergence problem of the whale optimization algorithm, the Lévy flight strategy [3] and the ranking-based mutation operator [20,27] are introduced into the whale optimization algorithm. The enhanced whale optimization algorithm can balance exploration and exploitation to find the global optimal solution.

Algorithm 1 Whale optimization algorithm

Begin
Step 1. Initialize the whale population $X_i (i = 1, 2, ..., n)$
Step 2. Calculate the fitness of each search agent
    Get the best search agent $\boldsymbol{X}^*$
Step 3. while ($t < t_{\max}$) do
        for each search agent
        Update parameters a, A, C, l and p
            if1 ($p < 0.5$)
                if2 ($|A| < 1$)
                    Update the position of the current search agent using Eq. (2)
                else if2 ($|A| \geq 1$)
                    Select a random search agent $\boldsymbol{X}_{rand}$
                    Update the position of the current search agent using Eq. (9)
                end if2
            else if1 ($p \geq 0.5$)
                    Update the position of the current search agent using Eq. (6)
            end if1
        end for
        Check if any search agent goes beyond the search space and amend it
        Calculate the fitness of each search agent
        Update $\boldsymbol{X}^*$ if there is a better solution
        $t = t + 1$
    end while
    return $\boldsymbol{X}^*$
End

3.1. Lévy flight strategy

The Lévy flight strategy is an effective random walk strategy that can expand the search space of the algorithm, avoid premature convergence and enhance the global search ability [3]. The Lévy flight strategy improves the calculation accuracy of the whale optimization algorithm. The position is calculated as follows:

$$\boldsymbol{X}(t+1) = \boldsymbol{X}(t) + \mu sign[rand - 1/2] \oplus Levy \tag{10}$$

where $\boldsymbol{X}$ is the position at t; μ is a uniformly distributed random number; $sign[rand - 1/2]$ has only three values, -1, 0 or 1; and $\oplus$ is entrywise multiplication.

The relationship between the step length of the Lévy flight and time t obeys the Lévy distribution. For the probability density function of the Lévy flight strategy, the position is calculated as follows:

$$Levy \sim u = t^{-\lambda}, \quad 1 < \lambda \leq 3 \tag{11}$$

where λ is a power coefficient. Mantegna's algorithm is applied to calculate the generated random step length of the Lévy flight strategy. The position is calculated as follows:

$$s = \frac{\mu}{|v|^{\frac{1}{\beta}}} \quad \mu \sim N(0, \sigma_\mu^2) \quad v \sim N(0, \sigma_v^2) \tag{12}$$

Algorithm 2 Ranking-based mutation operator of "DE/rand/1"

Begin
Sort the population, and assign the ranking and selection probability P_i for each wave
Randomly select $r_1 \in \{1, N_p\}$ {base vector index}
while $rand > p_{r_1}$ or $r_1 == i$
Randomly select $r_1 \in \{1, N_p\}$
end
Randomly select $r_2 \in \{1, N_p\}$ {terminal vector index}
while $rand > p_{r_2}$ or $r_2 == r_1$ or $r_2 == i$
Randomly select $r_2 \in \{1, N_p\}$
end
Randomly select $r_3 \in \{1, N_p\}$ {starting vector index}
while $r_3 == r_2$ or $r_3 == r_1$ or $r_3 == i$
Randomly select $r_3 \in \{1, N_p\}$
end
End

where s is a random step length, β is 1.5, and μ and v obey normal distributions. σ_u and σ_v are, respectively calculated as follows:

$$\sigma_u = \left[\frac{\Gamma(1+\beta) \cdot \sin(\pi\beta/2)}{\beta \cdot \Gamma[(1+\beta)/2] \cdot 2^{(\beta-1)/2}} \right]^{1/\beta} \qquad \sigma_v = 1 \tag{13}$$

where Γ is the standard gamma function.

3.2. Ranking-based mutation operator

The optimal selected search agents are sorted according to the fitness value of each search agent. That is, the population is sorted in ascending order (i.e., from best to worst) based on the fitness value of each wave. The ranking of search agents is calculated as follows:

$$R_i = N_p - i, \qquad i = 1, 2, \ldots, N_p \tag{14}$$

where N_p is the size of the population. The higher the ranking, the better the search agent. Each search agent is sorted, and selection probability P_i of the ith wave is calculated as follows:

$$p_i = \frac{R_i}{N_p}, \qquad i = 1, 2, \ldots, N_p \tag{15}$$

The ranking-based mutation operator "DE/rand/1" is given in Algorithm 2. Individuals with a higher ranking have a greater probability of being selected as the basis vector or terminal vector in the mutation operator, and the purpose is to spread the good information about the population to future generations. The ranking-based mutation operator is not based on the selection probability to select the starting vector [20,27]. If the two vectors of the differential vector are both selected from the higher ranked vectors, the search step size of the differential vector may decrease rapidly and lead to a local optimum.

The enhanced whale optimization algorithm has a strong global search ability and local search ability to find the global optimal solution. The enhanced whale optimization algorithm is given in Algorithm 3. A flowchart of the enhanced whale optimization algorithm is given in Fig. 2.

Algorithm 3 Enhanced whale optimization algorithm

Begin

Step 1. Initialize the whale population $X_i (i = 1, 2, ..., n)$

Step 2. Calculate the fitness of each search agent

Get the best search agent $\boldsymbol{X}^*$

Step 3. while ($t < t_{\max}$) do

for each search agent

Update parameters a , A , C , l and p

Sort the population, and assign the ranking and selection probability P_i for each search agent

/*ranking-based mutation stage*/

Randomly select $r_1 \in \{1, N_p\}$ {base vector index}

while $rand > p_{r_1}$ or $r_1 == i$

Randomly select $r_1 \in \{1, N_p\}$

end

Randomly select $r_2 \in \{1, N_p\}$ {terminal vector index}

while $rand > p_{r_2}$ or $r_2 == r_1$ or $r_2 == i$

Randomly select $r_2 \in \{1, N_p\}$

end

Randomly select $r_3 \in \{1, N_p\}$ {starting vector index}

while $r_3 == r_2$ or $r_3 == r_1$ or $r_3 == i$

Randomly select $r_3 \in \{1, N_p\}$

end /*end of ranking-based mutation stage*/

if1 ($p < 0.5$)

if2 ($|A| < 1$)

Update the position of the current search agent using Eq. (2)

else if2 ($|A| \geq 1$)

Select a random search agent $\boldsymbol{X}_{rand}$

Update the position of the current search agent using Eq. (9)

end if2

else if1 ($p \geq 0.5$)

Update the position of the current search agent using Eq. (6)

end if1

end for

for each search agent

Update the position of the current search agent based on the Lévy flight using Eq. (10).

end for

Check if any search agent goes beyond the search space and amend it

Calculate the fitness of each search agent

Update $\boldsymbol{X}^*$ if there is a better solution

$t = t + 1$

end while

return $\boldsymbol{X}^*$

End

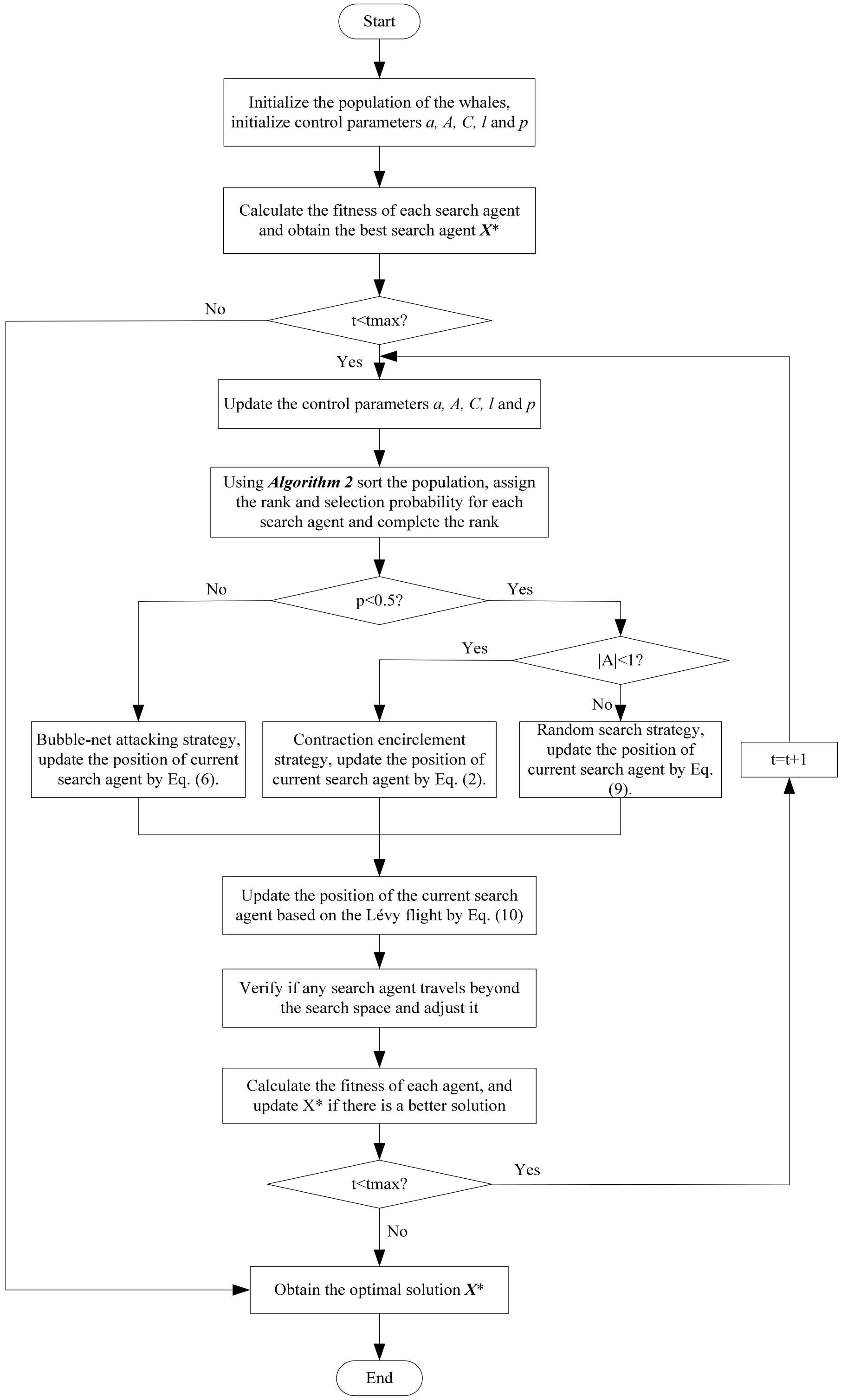


Fig. 2. Flowchart of the enhanced whale optimization algorithm.

3.3. Complexity analysis

In this section, both the time and space complexity of the proposed algorithm are analyzed.

3.3.1. Time complexity

The time complexity of the enhanced whale optimization algorithm is briefly discussed in this section. The enhanced whale optimization algorithm mainly includes six steps: initialization, ranking-based mutation, encircling the prey (exploration phase), bubble-net attacking the prey (exploitation phase), updating the whales' positions using the Lévy flight strategy and halting judgment. In the enhanced whale optimization algorithm, N denotes the population size, T denotes the maximum number of iterations, and D denotes the dimensionality of the problem. The time complexity of the enhanced whale optimization algorithm is given as follows: the initialization step has a double loop (N and D times), and the time complexity is $O(N * D)$. The ranking-based mutation stage has a triple loop (N, D and T times), and the time complexity is $O(N * D * T)$. For encircling the prey (exploration phase), bubble-net attacking the prey (exploitation phase) and updating the whales' positions using the Lévy flight strategy, there is a triple loop (N, D and T times), and the time complexity is $O(N * D * T)$. For halting judgment, the time complexity is $O(1)$. By analyzing all of the above steps, the total time complexity of the enhanced whale optimization algorithm is $O(N * D * T)$.

3.3.2. Space complexity

The space complexity of an algorithm is regarded as the storage space consumed by the algorithm. The population size is N and the dimensionality of the problem is D. The enhanced whale optimization algorithm is used to calculate the space complexity. The total space complexity of the enhanced whale optimization algorithm is $O(N * D)$. The whale optimization algorithm uses N search agents to calculate the space complexity, and the total space complexity of the whale optimization algorithm is $O(N * D)$. Therefore, the total space complexity of the whale optimization algorithm is the same as the total space complexity of the enhanced whale optimization algorithm, and the space efficiency of the enhanced whale optimization algorithm is effective and stable.

4. Simulation test and result analysis

4.1. Experimental setup

The numerical experiment is set up on a computer with an Intel Core i7-8750H 2.2 GHz CPU, a GTX1060, and 8 GB memory running on Windows 10.

4.2. Benchmark functions

To verify the overall optimization performance of the enhanced whale optimization algorithm, the proposed algorithm is used to solve the function optimization problem. Some information on the benchmark functions is given in Table 1. The benchmark functions are divided into three categories: $f_1 - f_6$ are the unimodal functions, $f_7 - f_{11}$ are the multimodal functions, and $f_{12} - f_{18}$ are the fixed-dimension multimodal functions.

The control parameters of all algorithms are representative empirical values, which are derived from the original articles. Some metaheuristic optimization algorithms include the artificial bee colony (ABC) [28], the cuckoo search (CS) [18], the dragonfly algorithm (DA) [47], moth–flame optimization (MFO) [46], particle swarm optimization (PSO) [10], water wave optimization (WWO) [77] and the whale optimization algorithm (WOA) [49]. The initial parameters for all algorithms are given in Table 2.

For each algorithm, the size of the population is 50, the maximum number of iterations is 1000, and the number of independent runs is 30. Best, Worst, Mean and Std are the optimal value, worst value, mean value and standard deviation, respectively. To highlight the effectiveness and feasibility of the enhanced whale optimization algorithm, the optimal value is shown in bold, and the ranking is based on the standard deviation.

In Table 3, for f_1 and f_3, the enhanced whale optimization algorithm can find the exact optimal value. The optimal value, worst value, mean value and standard deviation of the enhanced whale optimization algorithm are all zero, which indicates that the enhanced whale optimization algorithm has the ability to find the global optimal solution. For f_2, f_4 and f_6, the optimal value, worst value, mean value and standard deviation of the enhanced whale

Table 1
Benchmark functions.

Benchmark test functions	Dim	Range	$f_{\min}$
$f_1 = \sum_{i=1}^{n} x_i^2$	30	[−100,100]	0
$f_2(x) = \sum_{i=1}^{n} \|x_i\| + \prod_{i=1}^{n} \|x_i\|$	30	[−10,10]	0
$f_3(x) = \sum_{i=1}^{n} (\sum_{j=1}^{i} x_j)^2$	30	[−100,100]	0
$f_4(x) = \max_i \{\|x_i\|, 1 \le i \le D\}$	30	[−100,100]	0
$f_5(x) = \sum_{i=1}^{D-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	30	[−30,30]	0
$f_6(x) = \sum_{i=1}^{n} x_i^4 + random(0, 1)$	30	[−1.28,1.28]	0
$f_7(x) = \sum_{i=1}^{n} [x_i^2 - 10\cos(2\pi x_i) + 10]$	30	[−5.12,5.12]	0
$f_8(x) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2} - \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos 2\pi x_i\right)\right)$ $+20 + e$	30	[−32,32]	0
$f_9(x) = \frac{1}{4000}\sum_{i=1}^{n}\left(x_i^2\right) - \prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	30	[−600,600]	0
$f_{10}(x) = \frac{\pi}{D}\left\{10\sin^2(\pi y_1) + \sum_{i=1}^{D-1} \begin{matrix}(y-1)^2[1+10\sin^2(\pi y_1)]\\ +(y_D-1)^2\end{matrix}\right\}$ $+\sum_{i=1}^{D} u(x_i, 10, 100, 4)$ $y_i = 1 + \frac{x_i + 1}{4}$ $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m, x^i > a \\ 0, -a \le x_i \le a \\ k(-x_i - z)^m, x_i < a \end{cases}$	30	[−50,50]	0
$f_{11}(x) = 0.1\left\{\sin^2\left(3\pi x_1 + \sum_{i=1}^{n} \begin{matrix}(x_i-1)^2[1+\sin^2(3\pi x_i+1)]\\ +(x_n-1)^2[1+\sin^2(2\pi x_n)]\end{matrix}\right)\right\}$ $+\sum_{i=1}^{n} u(x_i, 5, 100, 4)$	30	[−50,50]	0
$f_{12}(x) = \sum_{i=1}^{11} [a_i - \frac{x_1(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4}]^2$	4	[−5,5]	0.0003075
$f_{13}(x) = -\frac{1 + \cos(12\sqrt{x_1^2 + x_2^2})}{0.5(x_1^2 + x_2^2) + 2}$	2	[−5.12,5.12]	−1
$f_{14}(x) = -\sum_{i=1}^{5} [(x - a_i)(x - a_i)^T + c_i]^{-1}$	4	[0,10]	−10.1532
$f_{15}(x) = -\sum_{i=1}^{7} [(x - a_i)(x - a_i)^T + c_i]^{-1}$	4	[0,10]	−10.4029

(*continued on next page*)

Table 1 (*continued*).

Benchmark test functions	Dim	Range	$f_{\min}$
$f_{16}(x) = -\sum_{i=1}^{10}[(x-a_i)(x-a_i)^T + c_i]^{-1}$	4	[0,10]	−10.5364
$f_{17}(x) = 0.5 + \frac{\sin^2\left(\sqrt{x_1^2+x_2^2}\right) - 0.5}{(1+0.001(x_1^2+x_2^2))^2}$	2	[−100,100]	−1
$f_{18}(x) = \sum_{i=1}^{n} x_i \sin(x_i) + 0.1x_i$	10	[−10,10]	0

Table 2
Initial parameters for all algorithms.

Algorithm	Parameters	Values
ABC [28]	Maximum number of searches $t_{\max}$	5D
CS [18]	Parameter β	1.5
	Probability P	0.25
DA [47]	Inertia weight ω	[0.2,0.9]
	Separation weight s	0.1
	Alignment weight a	0.1
	Cohesion weight c	0.7
	Food factor f	1
	Enemy factor e	1
MFO [46]	Constant b	1
	Random number t	[−1,1]
	Random number r	[−2,−1]
PSO [10]	Constant inertia ω	0.7298
	First acceleration coefficient c_1	1.4962
	Second acceleration coefficient c_2	1.4962
WWO [77]	Wavelength λ	0.5
	Wave height $h_{\max}$	12
	Wavelength reduction coefficient α	1.0026
	Breaking coefficient β	[0.01,0.25]
	Maximum number $k_{\max}$ of breaking directions	min(12, $D/2$)
WOA [49]	Random number r_1	[0,1]
	Random number r_2	[0,1]
	Convergence factor α	[0,2]
	Constant coefficient b	1
	Random number l	[−1,1]
EWOA [49], [3], [20] and [27]	Random number r_1	[0,1]
	Random number r_2	[0,1]
	Convergence factor α	[0,2]
	Constant coefficient b	1
	Random number l	[−1,1]
	Scaling factor F	0.7
	A power coefficient λ	(1,3]
	A random number β	1.5

optimization algorithm have been greatly improved compared to those of the basic whale optimization algorithm, and the related values of the enhanced whale optimization algorithm are better than those of other algorithms. The enhanced whale optimization algorithm is ranked first, which indicates that the enhanced whale optimization algorithm has strong stability. For f_5, the optimal value of the enhanced whale optimization algorithm is superior to those of the artificial bee colony, cuckoo search, dragonfly algorithm, moth–flame optimization algorithm, water wave optimization algorithm and whale optimization algorithm, but the standard deviation of the enhanced whale

Table 3
Simulation results for unimodal functions.

Function	Result	ABC	CS	DA	MFO	PSO	WWO	WOA	EWOA	Rank
f_1	Best	0.000441	0.003565	56.82441	5.40E−06	8.89E−21	4.78E−06	1.2E−187	**0**	1
	Worst	0.013106	0.018458	1223.146	10 000.00	2.10E−16	1.85E−05	1.7E−172	0	
	Mean	0.003955	0.008541	560.7034	666.6667	1.32E−17	8.69E−06	1.3E−173	0	
	Std	0.002820	0.003361	346.1621	2537.081	3.92E−17	3.65E−06	0	0	
f_2	Best	0.004953	0.580363	0.555639	8.76E−05	1.48E−10	0.071107	5.5E−120	**1.5E−261**	1
	Worst	0.026214	2.974981	23.25196	80.00000	3.75E−06	9.539024	5.9E−107	7.8E−223	
	Mean	0.013773	1.343994	11.04686	32.66745	1.47E−07	2.519594	2.1E−108	2.8E−224	
	Std	0.005029	0.552331	5.675405	22.58058	6.82E−07	2.538686	1.1E−107	0	
f_3	Best	18 084.41	238.1954	148.9677	207.2669	5.752056	292.7059	725.1963	**0**	1
	Worst	35 395.58	681.6806	23 417.78	43 342.69	149.9196	3501.594	27 662.65	0	
	Mean	28 287.66	453.0947	6964.777	17 452.92	38.25553	1685.305	10 251.24	0	
	Std	4990.142	105.6355	6670.608	12 382.95	36.61575	899.9194	6594.339	0	
f_4	Best	70.57095	2.061194	5.324946	43.52046	0.252266	1.787306	0.001564	**3.2E−250**	1
	Worst	87.76288	5.553504	37.84536	70.95349	3.314598	16.44062	85.56572	2.4E−200	
	Mean	81.04536	3.039861	12.92349	53.42016	1.269645	5.692033	29.23464	1.5E−201	
	Std	4.475353	0.799954	6.720430	6.950500	0.873020	4.226016	27.64749	0	
f_5	Best	82.45503	29.61354	1456.672	22.66173	**6.971996**	23.24552	26.08370	26.13592	2
	Worst	316.2463	82.00798	166 168.5	90 082.88	240.6298	368.8079	27.02721	27.96951	
	Mean	155.2698	41.22386	43 527.90	21 448.24	55.99213	70.78803	26.59918	26.79924	
	Std	50.01334	11.76614	47 193.08	38 511.77	51.09371	79.15149	0.26055	0.493509	
f_6	Best	0.163052	0.015381	0.013949	0.027481	0.168938	0.023333	2.82E−05	**1.58E−06**	1
	Worst	0.507113	0.053480	0.522144	26.88650	1.118292	0.110494	0.004864	0.000156	
	Mean	0.318867	0.031947	0.154740	3.395513	0.727117	0.052241	0.001194	4.42E−05	
	Std	0.079389	0.008405	0.117166	6.248611	0.275916	0.018929	0.00133	3.88E−05	

optimization algorithm is worse than that of the whale optimization algorithm and the is ranked second among all algorithms. The worst value and mean value of the enhanced whale optimization algorithm are relatively better than those of the other algorithms. The Lévy flight strategy increases the diversity of the population to avoid falling into a local optimum. The ranking-based mutation operator increases the probability of selecting the best search agent and enhances the local search ability. The enhanced whale optimization algorithm not only has strong robustness and stability but also has a faster convergence speed and higher calculation accuracy.

In Table 4, for f_7 and f_9, the enhanced whale optimization algorithm finds the exact optimal value. The worst value, mean value and standard deviation of the enhanced whale optimization algorithm are better than those of the other algorithms, which indicates that the enhanced whale optimization algorithm can avoid premature convergence to effectively solve the function optimization problem. Furthermore, the enhanced whale optimization algorithm based on the standard deviation is ranked first, which indicates that the enhanced whale optimization algorithm has strong stability and feasibility. For f_8, compared with the other algorithms, the optimal value of the enhanced whale optimization algorithm is the best. The worst value and mean value of the enhanced whale optimization algorithm not only have a great improvement but are also better than those of the other algorithms. The standard deviation of the enhanced whale optimization algorithm is the smallest of all algorithms, and the algorithm is ranked first. For f_{10}, the optimal value of the enhanced whale optimization algorithm is worse than that of the particle swarm optimization, but the worst value, mean value and standard deviation of the enhanced whale optimization algorithm are superior to those of the other algorithms except the artificial bee colony. For f_{11}, the optimal value of the particle swarm optimization is better than that of the enhanced whale optimization algorithm, but the standard deviation of the enhanced whale optimization algorithm is better than those of the cuckoo search, the dragonfly algorithm, moth–flame optimization and water wave optimization. The Lévy flight strategy has a strong global search ability, and the ranking-based mutation operator has a strong local search ability, so that the enhanced whale optimization algorithm can effectively balance the global search ability and the local search ability to find the optimal solution.

In Table 5, for f_{12}, the enhanced whale optimization algorithm finds the global optimal solution. The worst value, mean value and standard deviation of the whale optimization algorithm are better than those of other algorithms

Table 4
Simulation results for multimodal functions.

Function	Result	ABC	CS	DA	MFO	PSO	WWO	WOA	EWOA	Rank
f_7	Best	14.91939	64.75209	13.11588	47.75797	32.83361	31.24869	0	**0**	1
	Worst	39.08385	107.8317	188.8454	230.2609	106.4602	131.8193	0	0	
	Mean	25.68877	83.29308	107.8206	159.4580	57.17686	81.95623	0	0	
	Std	6.630500	9.587044	44.03340	46.58737	15.53451	25.09813	0	0	
f_8	Best	7.367225	2.346710	2.038453	0.001400	1.24E−10	2.221425	8.88E−16	**8.88E−16**	1
	Worst	15.03522	11.07992	11.55848	19.96300	2.738609	7.762519	7.99E−15	8.88E−16	
	Mean	9.555140	4.272634	6.558318	14.11526	0.758833	4.068076	3.97E−15	8.88E−16	
	Std	1.707736	1.696220	1.931670	8.643200	0.861509	1.263339	2.23E−15	0	
f_9	Best	0.004288	0.025822	1.151923	1.490E−05	0	5.31E−05	0	**0**	1
	Worst	0.146331	0.216970	15.09438	180.1740	0.075916	0.054089	0.098195	0	
	Mean	0.052684	0.092437	5.398671	12.04280	0.015305	0.011403	0.006107	0	
	Std	0.029505	0.049457	3.596906	39.13299	0.018947	0.012853	0.020521	0	
f_{10}	Best	7.73E−06	0.504269	0.357252	4.35E−06	**4.48E−20**	1.56236	0.00014	0.000412	2
	Worst	0.000702	1.909423	76.15341	2.56E+08	1.353868	6.319952	0.04473	0.013227	
	Mean	0.000124	1.133521	10.66097	8 533 334	0.214809	3.462968	0.004999	0.003329	
	Std	0.000138	0.330053	15.16098	46 738 995	0.382419	1.31019	0.010937	0.003566	
f_{11}	Best	0.000403	0.070825	4.479486	7.85E−06	**6.52E−19**	0.001075	0.004294	0.007623	4
	Worst	0.005361	0.395744	193 984.9	3.608714	0.010987	0.029491	0.225287	0.307367	
	Mean	0.001587	0.163772	13 444.39	0.258952	0.002197	0.008874	0.048533	0.042714	
	Std	0.001137	0.079455	39 111.95	0.753152	0.004470	0.007498	0.055836	0.054478	

except the cuckoo search. The ranking of the whale optimization algorithm is second. For f_{13} and f_{17}, the whale optimization algorithm can find the exact optimal value, and the worst value, mean value and standard deviation are all consistent. The whale optimization algorithm is ranked first. For f_{14} and f_{15}, the whale optimization algorithm and the other algorithms find accurate values. The worst value and mean value of the cuckoo search is better than those of the whale optimization algorithm, which means that the whale optimization algorithm has certain advantages in solving the fixed-dimension multimodal functions. The algorithm is ranked second. For f_{16}, all algorithms find the global optimal solution. The worst value, mean value and standard deviation of the whale optimization algorithm are better than those of the artificial bee colony, dragonfly algorithm, moth–flame optimization algorithm, particle swarm optimization algorithm and whale optimization algorithm. For f_{18}, the optimal value, worst value, mean value and standard deviation of the enhanced whale optimization algorithm are better than those of the other algorithms. The Lévy flight strategy can expand the search range of the algorithm to avoid search stagnation, and the ranking-based mutation operator enhances the optimization performance of the algorithm and accelerates convergence speed. The whale optimization algorithm achieves complementary advantages to enhance the global search ability so that the whale optimization algorithm can find an accurate solution.

The Wilcoxon rank-sum test is an important criterion to detect whether there is a significant difference between the whale optimization algorithm and the other algorithms [67]. A p-value greater than 0.05 in bold indicates that there is no significant difference. A p-value less than 0.05 indicates a significant difference. The results of the p-value Wilcoxon rank-sum test on the benchmark functions are given in Table 6.

Fig. 3 is the convergence curve of different algorithms when solving the function optimization problem. The convergence curve can effectively reflect the convergence speed and calculation accuracy of the whale algorithm. By comparing the convergence curve, we can intuitively see the overall optimization performance of each algorithm. For $f_1 - f_6$, the calculation accuracy of the enhanced whale optimization algorithm has been greatly improved compared to the basic whale optimization algorithm, as shown in Table 3. Furthermore, the optimal value of the enhanced whale optimization algorithm is better than those of other algorithms. Compared with other algorithms, the enhanced whale optimization algorithm has faster convergence and higher calculation accuracy. For $f_7 - f_{11}$, the enhanced whale optimization algorithm can increase the diversity of the population to jump out local optima, so the enhanced whale optimization algorithm has a strong global search ability for finding the better optimal value, as shown in Table 4. The convergence speed and calculation accuracy of the enhanced whale optimization algorithm are the best of all algorithms. For $f_{12} - f_{18}$, the enhanced whale optimization algorithm can find the global

Table 5
Simulation results for fixed-dimension multimodal functions.

Function	Result	ABC	CS	DA	MFO	PSO	WWO	WOA	EWOA	Rank
f_{12}	Best	0.000479	0.000307	0.000757	0.000311	0.000307	0.000308	0.000307	**0.000308**	2
	Worst	0.001525	0.000308	0.002252	0.001489	0.001091	0.001361	0.001490	0.000354	
	Mean	0.000970	0.000308	0.001342	0.000838	0.000780	0.000643	0.000629	0.000320	
	Std	0.000251	8.06E−08	0.000443	0.000339	0.000298	0.000196	0.000367	1.33E−05	
f_{13}	Best	**−1**	**−1**	**−1**	**−1**	**−1**	**−1**	**−1**	**−1**	1
	Worst	−0.99985	−1	−1	−0.93625	−1	−1	−0.93625	−1	
	Mean	−0.99998	−1	−1	−0.98300	−1	−1	−0.9915	−1	
	Std	4.03E−05	1.85E−12	1.84E−07	0.028675	0	0	0.022043	0	
f_{14}	Best	**−10.1532**	**−10.1532**	**−10.1532**	**−10.1532**	**−10.1532**	**−10.1532**	**−10.1532**	**−10.1532**	2
	Worst	−10.0516	−10.1532	−5.05454	−2.63047	−2.63047	−5.10077	−2.63041	−10.1353	
	Mean	−10.1476	−10.1532	−9.29915	−6.29876	−6.05499	−9.98478	−9.31083	−10.1481	
	Std	0.019748	3.35E−14	1.914104	3.136477	3.317015	0.922443	2.228984	0.004464	
f_{15}	Best	**−10.4029**	**−10.4029**	**−10.4029**	**−10.4029**	**−10.4029**	**−10.4029**	**−10.4029**	**−10.4029**	2
	Worst	−10.3731	−10.4029	−5.07499	−2.75193	−1.83759	−5.12882	−2.76589	−10.3765	
	Mean	−10.4003	−10.4029	−9.51504	−7.85513	−6.23857	−10.2271	−9.78701	−10.3964	
	Std	0.007603	1.63E−14	2.007579	3.449531	3.374958	0.962918	1.888867	0.005228	
f_{16}	Best	**−10.5364**	**−10.5364**	**−10.5364**	**−10.5364**	**−10.5364**	**−10.5364**	**−10.5364**	**−10.5364**	3
	Worst	−9.84074	−10.5364	−2.80663	−2.42173	−2.42173	−10.5364	−2.42164	−10.5232	
	Mean	−10.5110	−10.5364	−9.20326	−8.04077	−7.60607	−10.5364	−8.45949	−10.5325	
	Std	0.126670	3.62E−12	2.486978	3.378472	3.483117	2.58E−15	2.809569	0.003327	
f_{17}	Best	**−1**	**−1**	**−1**	**−1**	**−1**	**−1**	**−1**	**−1**	1
	Worst	−0.99028	−0.99996	−0.99028	−0.99028	−0.99028	−1	−0.99028	−1	
	Mean	−0.99628	−1	−0.99741	−0.99126	−0.99676	−1	−0.99449	−1	
	Std	0.003681	8.38E−06	0.00437	0.002965	0.004658	7.33E−08	0.004897	0	
f_{18}	Best	7.47E−05	0.061892	0.001964	2.24E−16	2.40E−32	1.68E−12	3.9E−121	**1.2E−276**	1
	Worst	0.001655	0.348609	4.849198	1.25E−14	6.38E−15	0.38795	3.919142	1.3E−216	
	Mean	0.000593	0.188827	0.789963	5.73E−15	1.50E−15	0.035615	0.198669	4.5E−218	
	Std	0.000373	0.058047	1.047417	3.10E−15	1.75E−15	0.087243	0.759318	0	

Table 6
Results of the *p*-value Wilcoxon rank-sum test on the benchmark functions.

Function	ABC	CS	DA	MFO	PSO	WWO	WOA
f_1	1.21E−12	1.21E−12	1.21E−12	1.21E−12	1.21E−12	1.21E−12	1.21E−12
f_2	3.02E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11
f_3	1.21E−12	1.21E−12	1.21E−12	1.21E−12	1.21E−12	1.21E−12	1.21E−12
f_4	3.02E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11
f_5	3.02E−11	3.02E−11	3.02E−11	1.11E−06	**9.71E−01**	3.01E−07	**1.09E−01**
f_6	3.02E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11	3.20E−09
f_7	1.21E−12	1.21E−12	1.21E−12	1.21E−12	1.21E−12	1.21E−12	N/A
f_8	1.21E−12	1.21E−12	1.21E−12	1.21E−12	1.19E−12	1.21E−12	9.16E−09
f_9	1.21E−12	1.21E−12	1.21E−12	1.21E−12	1.64E−08	1.21E−12	**8.15E−02**
f_{10}	6.07E−11	3.02E−11	3.02E−11	**5.20E−01**	**7.73E−02**	3.02E−11	**6.67E−03**
f_{11}	3.02E−11	6.72E−10	3.02E−11	2.92E−02	1.78E−10	1.16E−07	**6.84E−01**
f_{12}	3.02E−11	3.34E−11	3.02E−11	3.02E−11	3.99E−04	5.53E−08	6.01E−08
f_{13}	1.21E−12	1.93E−10	4.19E−02	2.15E−02	N/A	N/A	4.19E−02
f_{14}	6.71E−06	1.72E−11	3.91E−02	7.25E−02	**7.65E−02**	3.49E−10	2.13E−05
f_{15}	4.96E−07	2.05E−11	7.96E−03	6.97E−03	**7.68E−02**	3.13E−10	1.68E−04
f_{16}	8.53E−04	2.99E−11	2.92E−02	7.44E−03	**3.75E−01**	1.20E−11	**8.19E−01**
f_{17}	1.21E−12	1.21E−12	2.70E−03	8.99E−11	2.92E−04	**3.34E−01**	2.08E−06
f_{18}	3.02E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11

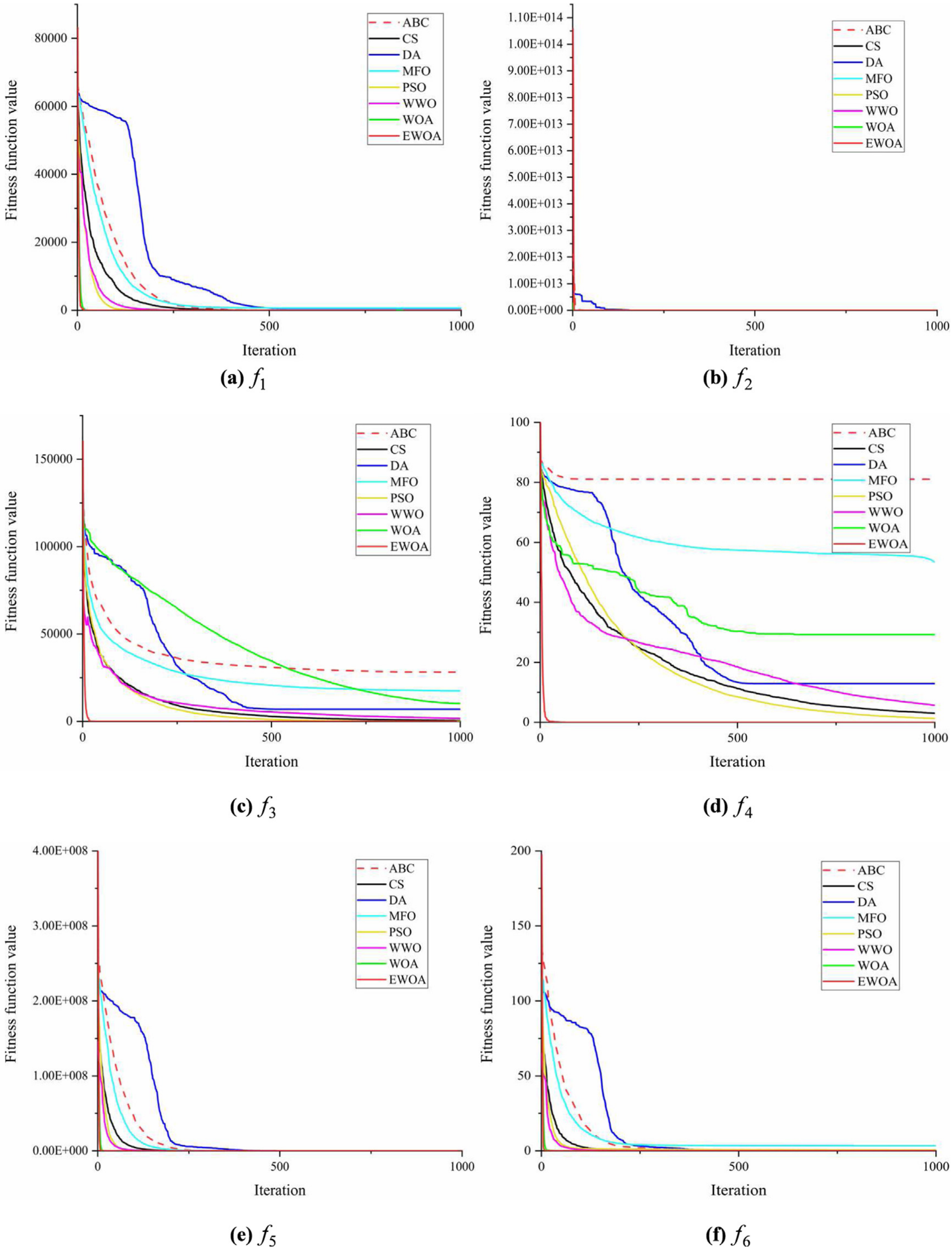
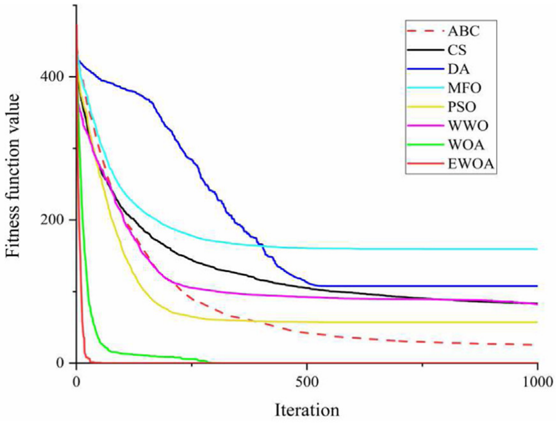
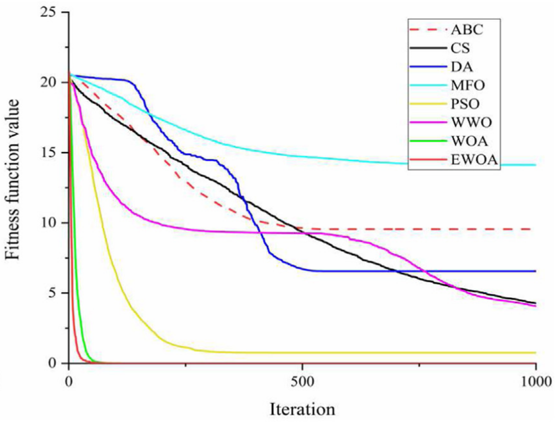


Fig. 3. Convergence curve of these algorithms when solving the benchmark functions.

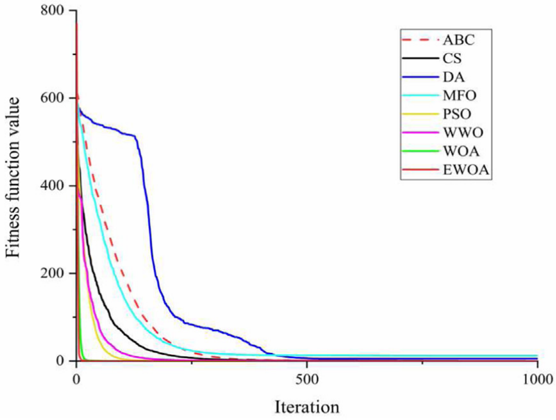
accurate solution, as shown in Table 5. The enhanced whale optimization algorithm can expand the search space and increase the probability of optimal individual selection to realize fixed-dimension multimodal functions. The overall



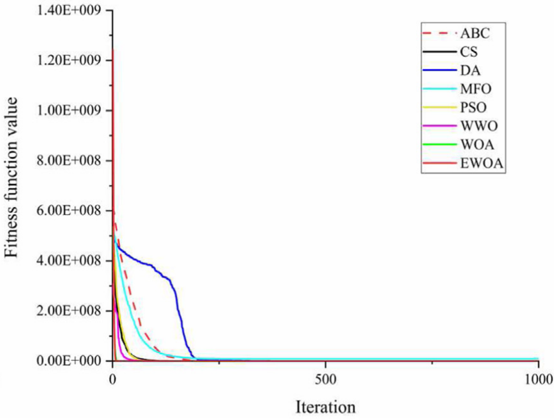
(g) f_7



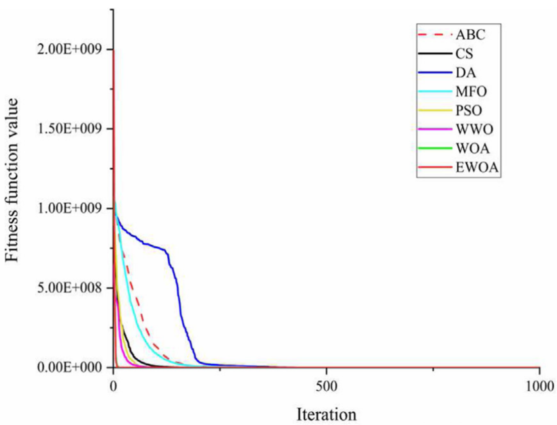
(h) f_8



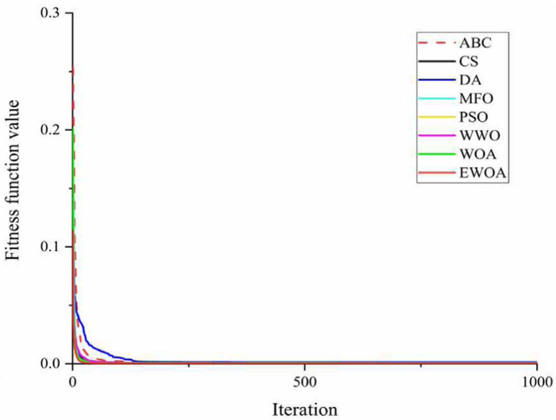
(i) f_9



(j) f_{10}



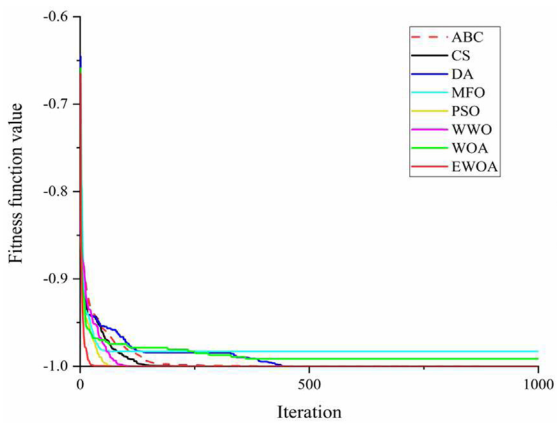
(k) f_{11}



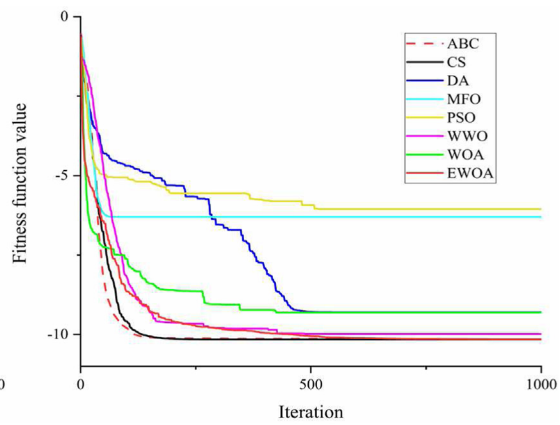
(l) f_{12}

Fig. 3. (*continued*).

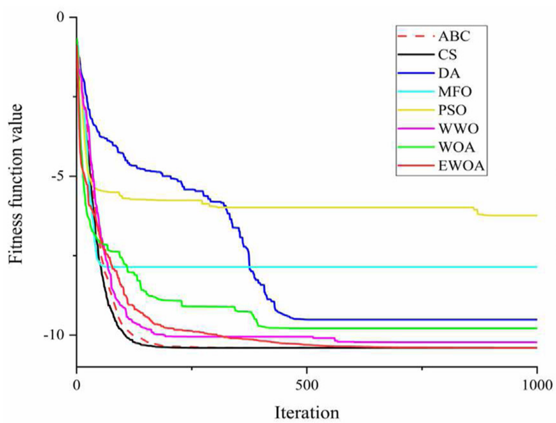
convergence effect of the enhanced whale optimization algorithm is better than those of the other algorithms. The Lévy flight strategy has a strong global search ability, which makes the algorithm jump out of local optima. The



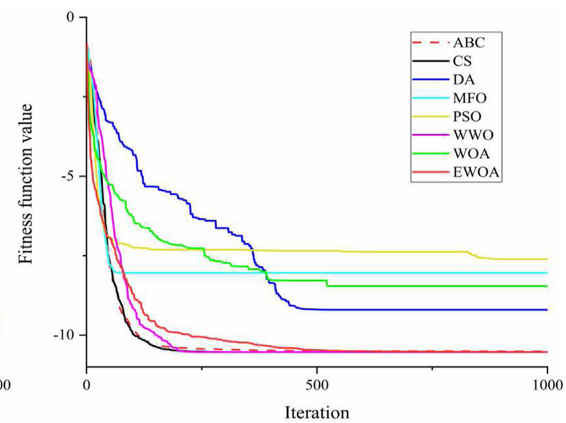
(m) f_{13}



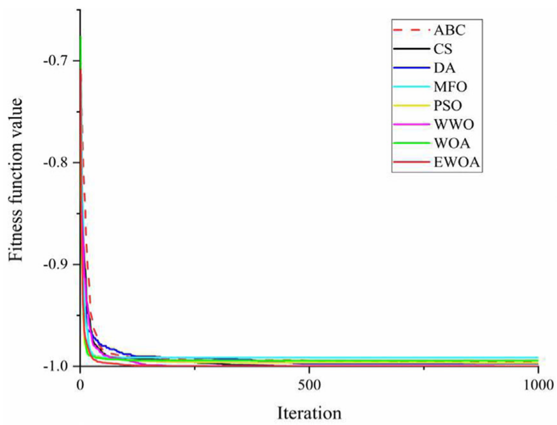
(n) f_{14}



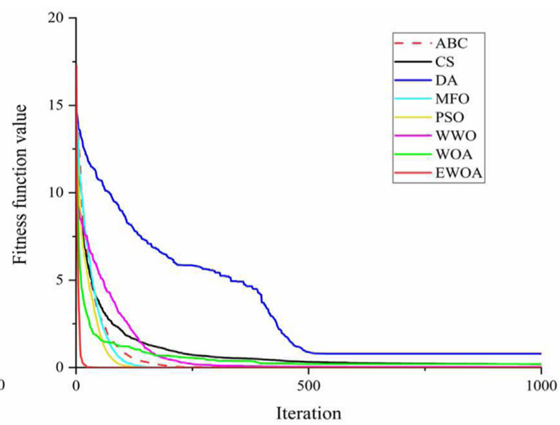
(o) f_{15}



(p) f_{16}



(q) f_{17}



(r) f_{18}

Fig. 3. (*continued*).

ranking-based mutation operator has a strong local search ability, which improves the convergence speed. These two important strategies are introduced into the basic whale optimization algorithm and can improve the robustness

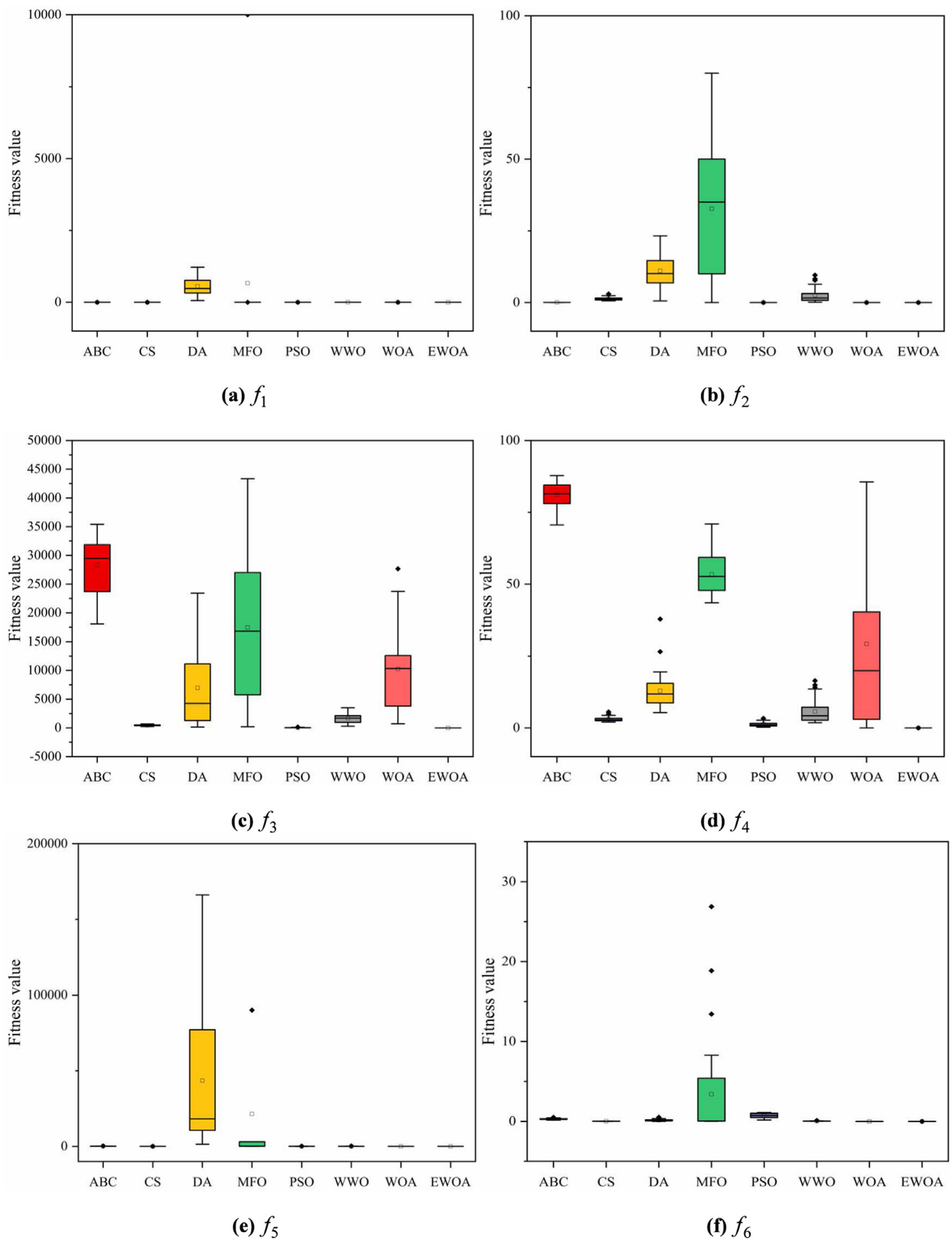
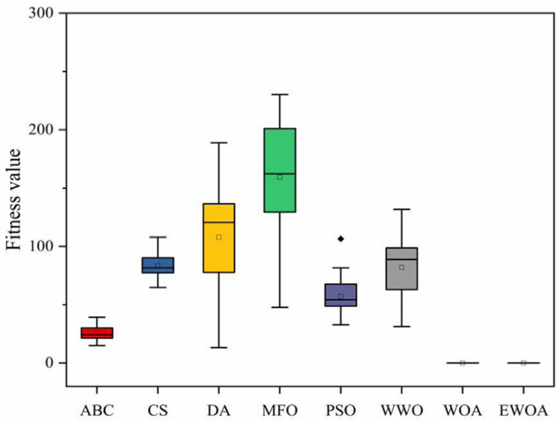
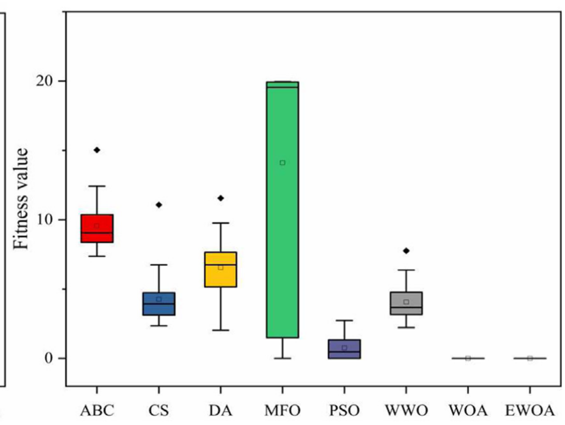


Fig. 4. ANOVA tests of these algorithms when solving the benchmark functions.

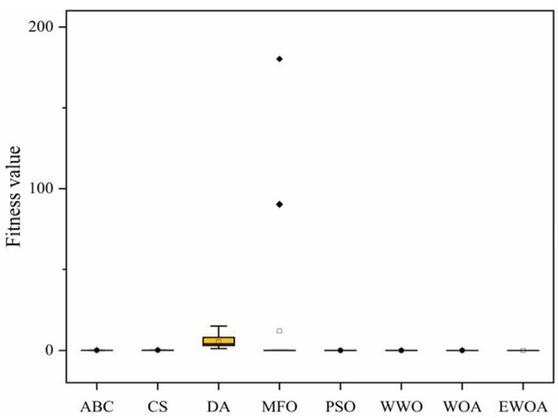
and stability of the algorithm. The enhanced whale optimization algorithm is an effective and feasible method for solving function optimization problems.



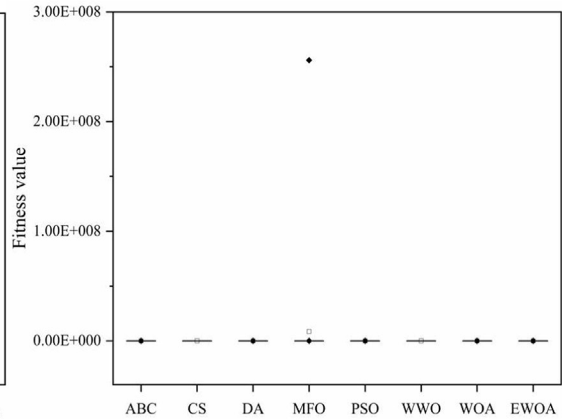
(g) f_7



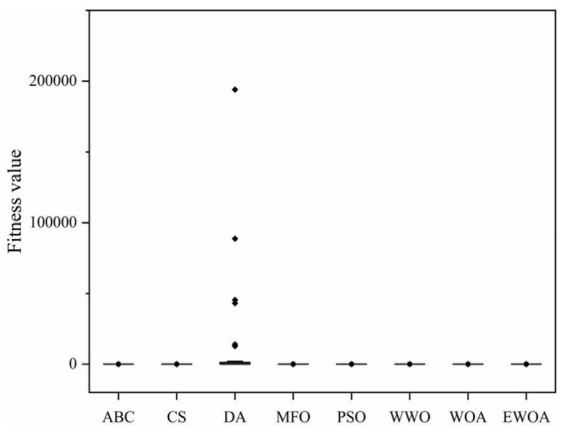
(h) f_8



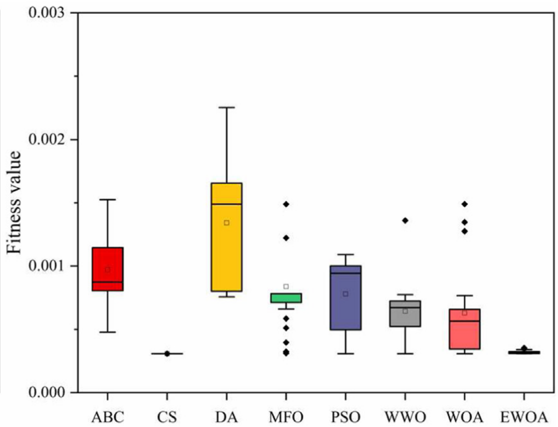
(i) f_9



(j) f_{10}



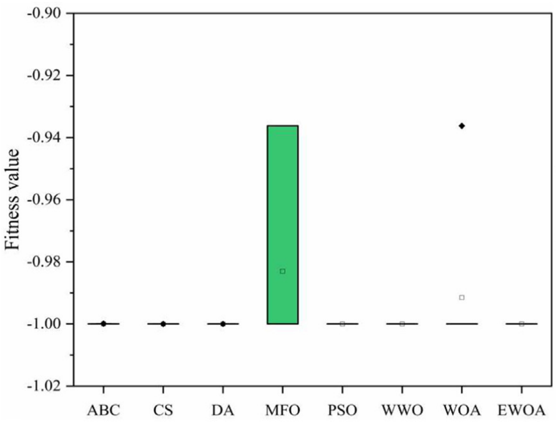
(k) f_{11}



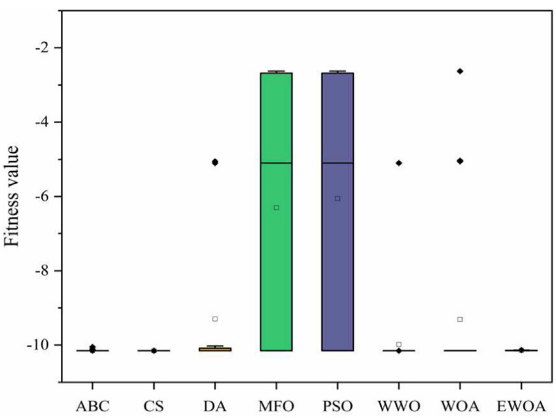
(l) f_{12}

Fig. 4. (*continued*).

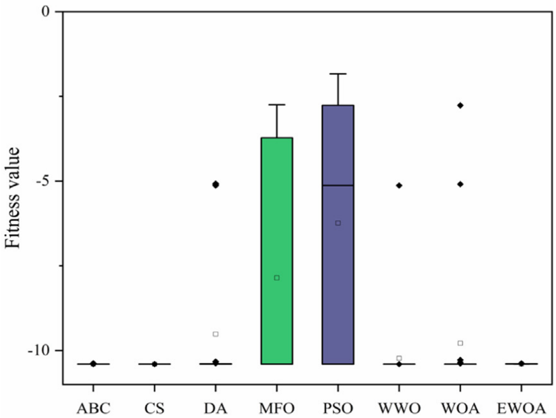
Fig. 4 shows the ANOVA test of different algorithms when solving the function optimization problems. The standard deviation can effectively reflect the stability of each algorithm; hence, we can intuitively see the stability and



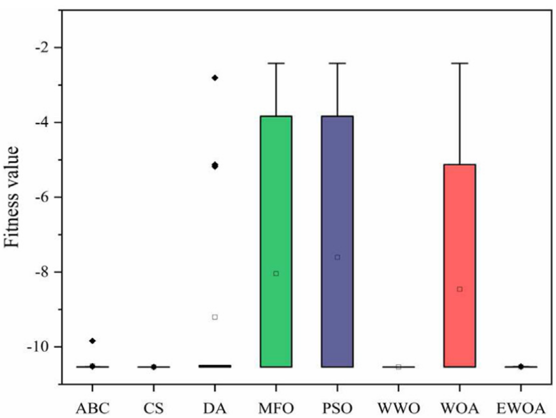
(m) f_{13}



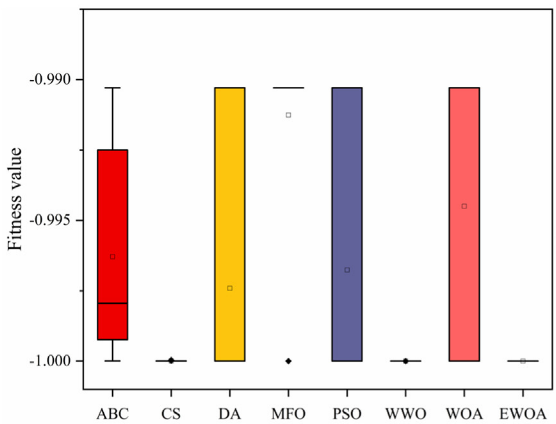
(n) f_{14}



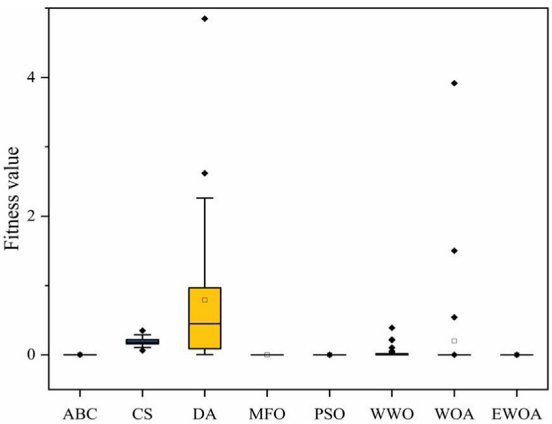
(o) f_{15}



(p) f_{16}



(q) f_{17}



(r) f_{18}

Fig. 4. (*continued*).

optimization performance. For $f_1 - f_6$, the standard deviation can effectively express the stability and feasibility of the algorithm. A smaller standard deviation indicates that an algorithm has better overall optimization performance. The standard deviation of the enhanced whale optimization algorithm is better than those of the other algorithms,

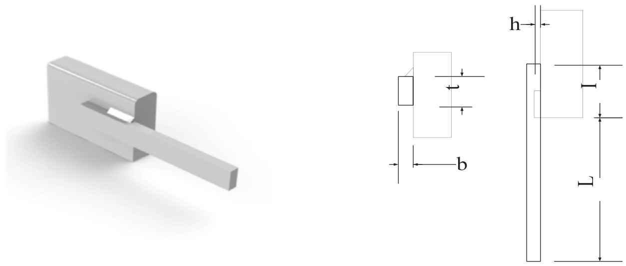


Fig. 5. Welded beam design problem.

and the enhanced whale optimization algorithm has the smallest standard deviation and strong stability. For $f_7 - f_{11}$, the basic whale optimization algorithm combines the advantages of the Lévy flight strategy and the ranking-based mutation operator so that the enhanced whale optimization algorithm has a strong global search ability and local search ability to obtain higher calculation accuracy. Compared with the basic whale optimization algorithm, the optimization result of the enhanced whale optimization algorithm has been greatly improved. The standard deviation of the enhanced whale optimization algorithm is relatively small, and the enhanced whale optimization algorithm has a certain stability. For $f_{12} - f_{18}$, the enhanced whale optimization algorithm can expand the search range and avoid search stagnation in order to find an accurate global solution. The enhanced whale optimization algorithm has better convergence speed and calculation accuracy. The standard deviation of the enhanced whale optimization algorithm is smaller than those of the other algorithms. The optimization result and standard deviation of the enhanced whale optimization algorithm are significantly superior to those of the other algorithms. This shows that the enhanced whale optimization algorithm has strong stability and robustness.

4.3. Enhanced whale optimization algorithm for solving structure engineering design problems

To verify the effectiveness and feasibility of the enhanced whale optimization algorithm, the proposed algorithm is applied to solve the following engineering design problems: the welded beam design problem [62], tension/compression spring design problem [39], pressure vessel design problem [76], cantilever beam design problem [45] and speed reducer design problem [79].

4.3.1. Welded beam design problem

The welded beam design problem is a classic structural engineering design problem [62]. The purpose is to minimize the manufacturing cost. As shown in Fig. 5, some important constraints are as follows: shear stress (τ), bending stress in the beam (σ), end deflection of the beam (δ), buckling load on the bar (P_c), and side constraints. There are four control variables: thickness of the weld (h), length of the clamped bar (l), width of the bar (t), and thickness of the bar (b). The mathematical model of the welded beam design problem is as follows:

Consider

$$x = [x_1 \ x_2 \ x_3 \ x_4] = [h \ l \ t \ b] \tag{16}$$

Minimize

$$f(x) = 1.10471 x_1^2 x_2 + 0.04811 x_3 x_4 (14.0 + x_2) \tag{17}$$

Subject to

$$g_1(x) = \tau(x) - \tau_{\max} \le 0 \tag{18}$$

$$g_2(x) = \sigma(x) - \sigma_{\max} \le 0 \tag{19}$$

$$g_3(x) = \delta(x) - \delta_{\max} \le 0 \tag{20}$$

$$g_4(x) = x_3 - x_4 \le 0 \tag{21}$$

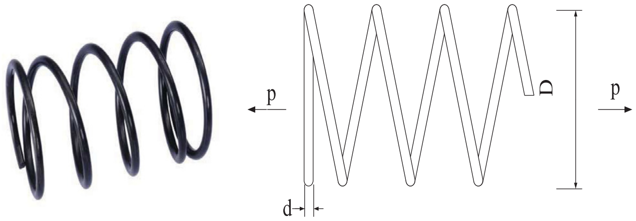


Fig. 6. Tension/compression spring design problem.

$$g_5(x) = P - P_c(x) \leq 0 \tag{22}$$

$$g_6(x) = 0.125 - x_1 \leq 0 \tag{23}$$

$$g_7(x) = 1.1047x_1^2 + 0.04811x_3x_4(14.0 + x_2) - 5.0 \leq 0 \tag{24}$$

$$\text{Variable range} \quad 0.1 \leq x_1 \leq 2; \quad 0.1 \leq x_2 \leq 10; \quad 0.1 \leq x_3 \leq 10; \quad 0.1 \leq x_4 \leq 2 \tag{25}$$

$$\text{where} \quad \tau(x) = \sqrt{(\tau')^2 + 2\tau'\tau''\frac{x_2}{2R} + (\tau'')^2} \tag{26}$$

$$\tau' = \frac{P}{\sqrt{2}x_x x_2}, \quad \tau'' = \frac{MP}{J}, \quad M = P(L + \frac{X_2}{2}) \tag{27}$$

$$R = \sqrt{\frac{x_2^2}{4} + (\frac{x_1 + x_3}{2})^2} \tag{28}$$

$$J = 2\left\{\sqrt{2}x_1x_2\left[\frac{x_2^2}{4} + (\frac{x_1 + x_3}{2})^2\right]\right\} \tag{29}$$

$$\sigma(x) = \frac{6PL}{x_4x_3^2}, \quad \delta(x) = \frac{6PL}{Ex_3^2x_4} \tag{30}$$

$$P_c(x) = \frac{4.103E\sqrt{\frac{x_3^2x_4^6}{36}}}{L^2}\left(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right) \tag{31}$$

The comparison results of the welded beam design problem are given in Table 7. Compared with other algorithms, the enhanced whale optimization algorithm can find better control parameter values and better objective function values. The results show that the enhanced whale optimization algorithm has a stronger global search ability and better optimization performance when used to solve the welded beam design problem, and the algorithm saves engineering design costs.

4.3.2. Tension/compression spring design problem

The tension/compression spring design problem is a broader engineering design problem [39]. The purpose is to minimize the weight of a tension/compression spring. As shown in Fig. 6, some necessary restrictions are as follows: minimum deflection (g_1), shear stress (g_2), surge frequency (g_3), and limits on the outside diameter (g_4). There are three control variables: the wire diameter (d), mean coil diameter (D), and number of active coils (N). The mathematical model of the tension/compression spring design problem is as follows:

Consider

$$x = [x_1 \ x_2 \ x_3] = [d \ D \ N] \tag{32}$$

Minimize

$$f(x) = (x_3 + 2)x_2x_1^2 \tag{33}$$

Table 7
Comparison results of the welded beam design problem.

Algorithm	Optimal values for variables				Optimal cost
	h	l	t	b	
GWO [51]	0.205676	3.478377	9.03681	0.205778	1.72624
GSA [51]	0.182129	3.856979	10.0000	0.202376	1.87995
CPSO [51]	0.202369	3.544214	9.048210	0.205723	1.72802
GA (Coello) [8]	N/A	N/A	N/A	N/A	1.8245
GA (Deb) [12]	N/A	N/A	N/A	N/A	2.3800
GA (Deb) [11]	0.2489	6.1730	8.1789	0.2533	2.4331
HS (Lee and Geem) [32]	0.2442	6.2231	8.2915	0.2443	2.3807
Random [57]	0.4575	4.7313	5.0853	0.6600	4.1185
Simplex [57]	0.2792	5.6256	7.7512	0.2796	2.5307
David [57]	0.2434	6.2552	8.2915	0.2444	2.3841
Approx [57]	0.2444	6.2189	8.2915	0.2444	2.3815
BA [70]	0.2015	3.562	9.0414	0.2057	1.7312065
CDE [25]	0.20317	3.542998	9.033498	0.206179	1.733462
WCA [15]	0.205728	3.470522	9.036620	0.205729	1.724856
MBA [61]	0.205729	3.470493	9.036626	0.205729	1.724853
IACO [30]	0.205700	3.471131	9.036683	0.205731	1.724918
RO [29]	0.203687	3.528467	9.004263	0.207241	1.735344
SaC [60]	0.244438	6.237967	8.288576	0.244566	2.3854347
PSO-DE [35]	N/A	N/A	N/A	N/A	1.7248531
WSA [4]	0.2057296	3.4704899	9.0366239	0.2057296	1.72485254
EOMSA [39]	0.22425	3.2486	8.6518	0.22445	1.7246
EWOA	0.2053	3.2652	9.0231	0.20811	1.7118

Subject to

$$g_1(x) = 1 - \frac{x_2^3 x_3}{71\,785 x_1^4} \le 0 \tag{34}$$

$$g_2(x) = \frac{4x_2^2 - x_1 x_2}{12\,566(x_2 x_1^3 - x_1^4)} + \frac{1}{5108 x_1^2} \le 0 \tag{35}$$

$$g_3(x) = 1 - \frac{140.45 x_1}{x_2^2 x_3} \le 0 \tag{36}$$

$$g_4(x) = \frac{x_1 + x_2}{1.5} - 1 \le 0 \tag{37}$$

$$\text{Variable range} \quad 0.05 \le x_1 \le 2.00; \quad 0.25 \le x_2 \le 1.30; \quad 2.00 \le x_3 \le 15.0 \tag{38}$$

The comparison results of the tension/compression spring design problem are given in Table 8. The optimal cost found by the enhanced whale optimization algorithm is the smallest of all algorithms, and the control parameters of the enhanced whale optimization algorithm are relatively good. The results show that the enhanced whale optimization algorithm has higher convergence accuracy when finding the global optimal solution, which not only reduces the cost of the spring design problem but also improves the optimization efficiency of the algorithm.

4.3.3. Pressure vessel design problem

The pressure vessel design problem is a classic mixed optimization constraint problem [76]. The purpose is to minimize the total costs of the materials, forming and welding. As shown in Fig. 7, some design parameters are as follows: the thickness of the shell (T_s), thickness of the head (T_h), inner radius (R), and length of the cylindrical section of the vessel (L). The mathematical model of the pressure vessel design problem is as follows:

Consider

$$x = [x_1 \ x_2 \ x_3 \ x_4] = [T_s \ T_h \ R \ L] \tag{39}$$

Table 8
Comparison results of the tension/compression spring design problem.

Algorithm	Optimal values for variables			Optimal cost
	d	D	N	
RO [29]	0.051370	0.349096	11.76279	0.0126788
SaC [60]	0.0521602	0.368158695	10.6484422	0.012669249
BGRA [24]	0.0516747	0.3563726	1.309229	0.012665237
IHS [40]	0.0511543	0.3498711	12.0764321	0.0126706
NM-PSO [72]	0.051620	0.355498	11.3333272	0.0126706
CPSO [21]	0.051728	0.357644	11.24454	0.0126747
YYPO [55]	0.051705	0.35710	11.266	0.12665
GA (Coello) [8]	0.051480	0.351661	11.632201	0.0127048
ES [42]	0.051643	0.355360	11.397926	0.012698
UPSO [21]	N/A	N/A	N/A	0.01312
CDE [25]	0.051609	0.354714	11.410831	0.0126702
ABC [1]	0.051749	0.358179	11.203763	0.012665
MFO [46]	0.051994457	0.36410932	10.868421862	0.0126669
GWO [51]	0.05169	0.356737	11.28885	0.012666
AFA [5]	0.0516674837	0.3561976945	11.3195613646	0.0126653049
BA [70]	0.05169	0.35673	11.2885	0.01267
LSA-SM [37]	0.05170453	0.3570899	11.26718	0.01266524
EWOA	0.0516772	0.356089	11.27684	0.0126649

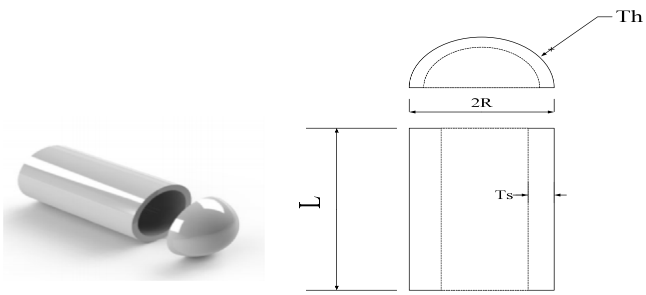


Fig. 7. Pressure vessel design problem.

Minimize

$$f(x) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.8x_1^2x_3 \tag{40}$$

Subject to

$$g_1(x) = -x_1 + 0.0193x_3 \le 0 \tag{41}$$
$$g_2(x) = -x_3 + 0.00954x_3 \le 0 \tag{42}$$
$$g_3(x) = -\pi x_3^2x_4 - \frac{4}{3}\pi x_3^3 + 1\,296\,000 \le 0 \tag{43}$$
$$g_4(x) = x_4 - 240 \le 0 \tag{44}$$

$$\text{Variable range } 0 \le x_1 \le 99; \quad 0 \le x_2 \le 99; \quad 10 \le x_3 \le 200; \quad 10 \le x_4 \le 200 \tag{45}$$

The comparison results of the pressure vessel design problem are given in Table 9. The optimal cost of the enhanced whale optimization algorithm has been greatly improved compared to those of the other algorithms. The enhanced whale optimization algorithm finds the global optimal solution according to the optimal control parameters. The results show that the enhanced whale optimization algorithm balances exploration and exploitation to enhance

Table 9
Comparison results of the pressure vessel design problem.

Algorithm	Optimal values for variables				Optimal cost
	T_s	T_h	R	L	
GSA [59]	1.125	0.625	55.9886598	84.4542025	8538.8359
GA [52]	0.9375	0.5	48.329	112.679	6410.3811
WEO [11]	0.8125	0.4375	42.0984	176.6366	6059.7143
PSO-GA [41]	0.7781686	0.3846491	40.3196187	200	5885.3327736
Lagrangian multiplier [48]	1.125	0.625	58.291	43.69	7198.0428
Branch-bound [19]	1.125	0.625	47.7	117.701	8129.1036
EOMSA [39]	1.14602	0.566477	59.3791	37.8283	5879.7727
GA (Coello) [8]	0.8125	0.4375	40.3239	200	6228.7445
CPSO [21]	0.8125	0.4375	42.091266	176.7465	6061.0777
CDE [25]	0.8125	0.4375	42.098411	176.7465	6059.7340
ABC [1]	0.8125	0.4375	42.098446	176.636596	6059.714339
BA [70]	0.8125	0.4375	42.0984456	176.6365958	6059.7143348
AFA [5]	0.8125	0.4375	42.09844611	176.6365894	6059.7142719
ACO [31]	0.8125	0.4375	42.103624	176.572656	6059.0888
ES [42]	0.8125	0.4375	42.098087	176.640518	6059.7456
MFO [46]	0.8125	0.4375	42.098445	176.636596	6059.7143
TLBO [58]	N/A	N/A	N/A	N/A	6059.714335
LSA-SM [37]	0.8103764	0.4005695	41.98842	178.0048	5942.6966
HHO [22]	0.8175838	0.4072927	42.09174576	176.7196352	6000.46259
BIANCA [54]	0.8125	0.4375	42.0968	176.658	6059.9384
MDDE [43]	0.8125	0.4375	42.0968446	176.636047	6059.70166
EWOA	0.8112138	0.4248752	42.08079	176.8759	5862.3411

the overall optimization performance. The enhanced whale optimization algorithm has strong robustness and high optimization efficiency when solving the pressure vessel design problem.

4.3.4. Cantilever beam design problem

The cantilever beam design problem is composed of five hollow squares, and each section is square. The purpose is to minimize the weight of the cantilever beam [45]. As shown in Fig. 8, the cantilever beam design problem has five control parameters, and the mathematical model is as follows:

Consider

$$x = [x_1\ x_2\ x_3\ x_4\ x_5] \tag{46}$$

Minimize

$$f(x) = 0.6224(x_1 + x_2 + x_3 + x_4 + x_5) \tag{47}$$

Subject to

$$g(x) = \frac{61}{x_1^3} + \frac{37}{x_2^3} + \frac{19}{x_3^3} + \frac{7}{x_4^3} + \frac{1}{x_5^3} \leq 1 \tag{48}$$

$$\text{Variable range } 0.01 \leq x_1 \leq 100;\ 0.01 \leq x_2 \leq 100;\ 0.01 \leq x_3 \leq 100;\ 0.01 \leq x_4 \leq 100;\ 0.01 \leq x_5 \leq 100 \tag{49}$$

The comparison results of the cantilever beam design problem are given in Table 10. The control parameters and optimal value of the enhanced whale optimization algorithm are obviously better than those of the other algorithms. The results show that the enhanced whale optimization algorithm can expand the search range to avoid falling into a local optimum; hence, the enhanced whale optimization algorithm has strong stability and feasibility for solving the cantilever beam design problem.

Table 10
Comparison results of the cantilever beam design problem.

Algorithm	Optimal value for variables					Optimal cost
	x_1	x_2	x_3	x_4	x_5	
MMA [9]	6.01	5.3	4.49	3.49	2.15	1.34
GCA_I [9]	6.01	5.3	4.49	3.49	2.15	1.34
GCA_II [9]	6.01	5.3	4.49	3.49	2.15	1.34
CS [18]	6.0089	5.3049	4.5023	3.5077	2.1504	1.33999
SOS [50]	6.01878	5.30344	4.49587	3.49896	2.15564	1.33996
MVO [50]	6.02394	5.30601	4.49501	3.49602	2.15273	1.3399595
ELPSO [78]	6.016	5.3092	4.4943	3.5015	2.1527	1.34
EWOA	6.01867	5.31481	4.49132	3.49907	2.15234	1.3399591

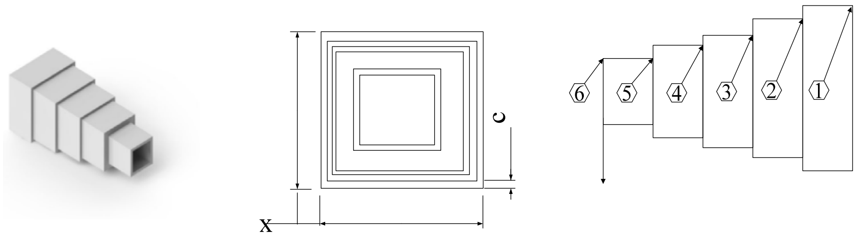


Fig. 8. Cantilever beam design problem.

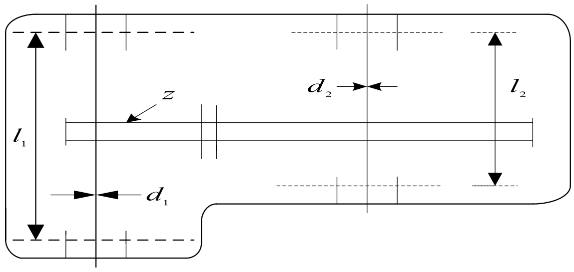


Fig. 9. Speed reducer design problem.

4.3.5. Speed reducer design problem

The speed reducer design problem is a typical complex multiconstraint optimization problem [79]. The purpose is to minimize the weight of the speed reducer. As shown in Fig. 9, the control variables are as follows: the width (b), module of teeth (m), number of teeth in the pinion (z), length of the first shaft between the bearings (l_1), length of the second shaft between bearings (l_2), diameter of the first shaft (d_1), and diameter of the second shaft (d_2). The mathematical model of the speed reducer design problem is as follows:

Consider

$$x = [x_1\ x_2\ x_3\ x_4\ x_5\ x_6\ x_7] = [b\ m\ z\ l_1\ l_2\ d_1\ d_2] \tag{50}$$

Minimize

$$\begin{aligned} f(x) = {} & 0.7854x_1x_2^2(3.3333x_3^2 + 14.9334x_3 - 43.0934) \\ & -1.508x_1(x_6^2 + x_7^2) + 7.4777(x_6^3 + x_7^3) + 0.7854(x_4x_6^2 + x_5x_7^2) \end{aligned} \tag{51}$$

Subject to

$$g_1(x) = -x_1 + 0.0193x_3 \le 0 \tag{52}$$

Table 11
Comparison results of the speed reducer design problem.

Algorithm	Optimal values for variables							Optimal cost
	b	m	z	l_1	l_2	d_1	d_1	
ABC [1]	3.5	0.7	17	7.3	7.715319	3.350214	5.286654	2994.471066
CS [18]	3.5	0.7	17	7.3	7.715319	3.350214	5.286654	2994.471066
HCPS [36]	3.500022	0.7	17.000012	7.300427	7.715377	3.350230	5.286663	2994.499107
SCA [60]	0.350001	0.7	17	7.300156	7.800027	3.350221	5.286685	2996.356689
LMFO [33]	3.5	0.7	17	7.3	7.8	3.350214	5.2866832	2996.348167
MBA [61]	3.5	0.7	17	7.300033	7.715772	3.350218	5.286654	2994.482453
MBFPA [66]	3.5	0.7	17	7.3	7.7153199122	3.35021466	5.28665446	2994.341315
WCA [35]	3.5	0.7	17	7.3	7.715319	3.350214	5.286654	2994.471066
PSODE [35]	3.5	0.7	17	7.3	7.8	3.350214	5.2866832	2996.348167
MDE [44]	3.50001	0.7	17	7.300156	7.800027	3.350221	5.286685	2996.356689
HEAA [65]	3.500022	0.7	17.000012	7.300427	7.715377	3.35023	5.286663	2994.499107
PVS [63]	3.49999	0.6999	17	7.3	7.8	3.3502	5.2866	2996.3481
EWOA	3.51063	0.7	17	7.3	7.8	3.35908	5.28998	2993.7658

$$g_2(x) = \frac{397.5}{x_1 x_2^2 x_3} - 1 \le 0 \tag{53}$$

$$g_3(x) = \frac{1.93 x_4^3}{x_2 x_6^4 x_3} - 1 \le 0 \tag{54}$$

$$g_4(x) = \frac{1.93 x_5^3}{x_2 x_7^5 x_3} - 1 \le 0 \tag{55}$$

$$g_5(x) = \frac{[(745 x_4 / x_2 x_3)^2 + 16.9 \times 10^6]^{1/2}}{110 x_6^3} - 1 \le 0 \tag{56}$$

$$g_6(x) = \frac{[(745 x_5 / x_2 x_3)^2 + 157.5 \times 10^6]^{1/2}}{85 x_7^3} - 1 \le 0 \tag{57}$$

$$g_7(x) = \frac{x_2 x_3}{40} - 1 \le 0 \tag{58}$$

$$g_8(x) = \frac{5 x_2}{x_1} - 1 \le 0 \tag{59}$$

$$g_9(x) = \frac{x_1}{12 x_2} - 1 \le 0 \tag{60}$$

$$g_{10}(x) = \frac{1.5 x_6 + 1.9}{x_4} - 1 \le 0 \tag{61}$$

$$g_{11}(x) = \frac{1.1 x_7 + 1.7}{x_5} - 1 \le 0 \tag{62}$$

$$\text{Variable range} \quad 2.6 \le x_1 \le 3.6;\ 0.7 \le x_2 \le 0.8;\ 17 \le x_3 \le 28;\ 7.3 \le x_4 \le 8.3;\ 7.3 \le x_5 \le 8.3;$$
$$2.9 \le x_6 \le 3.9;\ 5.0 \le x_7 \le 5.5 \tag{63}$$

The comparison results of the speed reducer design problem are given in Table 11. Compared with the other algorithms, the enhanced whale optimization algorithm has better calculation accuracy. That is, the enhanced whale optimization algorithm finds the optimal function value and control parameters. The results show that the enhanced whale optimization algorithm combines the advantages of the Lévy flight strategy and the ranking-based mutation operator to overcome premature convergence. The enhanced whale optimization algorithm is a good optimization method for solving the speed reducer design problem.

5. Conclusions and future research

In this paper, an enhanced whale optimization algorithm based on the Lévy flight strategy and the ranking-based mutation operator is proposed to solve the function optimization and structural engineering design problems. The purpose of the optimization is to find the global optimal solution and the minimum optimization cost. The Lévy flight strategy has a strong global search ability, which improves the calculation accuracy of the whale optimization algorithm. The ranking-based mutation operator has a strong local search ability, which accelerates the convergence speed. Therefore, the enhanced whale optimization algorithm can arbitrarily switch the global search ability and local search ability to obtain better optimization results. For function optimization, the enhanced whale optimization algorithm not only expands the search space to avoid premature convergence but also increases the probability of optimal individual selection to enhance the optimization ability. The optimization efficiency of the enhanced whale optimization algorithm is significantly better than those of the other algorithms, and the calculation accuracy of the enhanced whale optimization algorithm has been greatly improved compared to the basic whale optimization algorithm. The enhanced whale optimization algorithm obtains a smaller standard deviation, which indicates that the enhanced whale optimization algorithm has strong stability. For structural engineering design problems, the enhanced whale optimization algorithm has certain advantages in finding global solutions. Compared with other algorithms, the enhanced whale optimization algorithm not only minimizes the cost of engineering design problems but also has better optimization efficiency. The experimental results show that the overall optimization performance of the enhanced whale optimization algorithm is superior to those of the other algorithms. The convergence speed and the calculation accuracy of the enhanced whale optimization algorithm are better than those of the other algorithms. Furthermore, the enhanced whale optimization algorithm has strong practicability and feasibility for solving optimization problems.

In future research, the enhanced whale optimization algorithm based on Kapur's entropy method will be applied to solve the underwater multilevel thresholding image segmentation problem. The enhanced whale optimization algorithm will be compared with other algorithms by maximizing the fitness value of Kapur's entropy to find the best threshold values. The fitness value, peak signal-to-noise ratio (PSNR), structure similarity index (SSIM), execution time and Wilcoxon's rank-sum test will verify the segmentation performance and effectiveness of the enhanced whale optimization algorithm. The enhanced whale optimization algorithm will be used to solve autonomous underwater vehicle path planning problem. The optimization algorithm will enable an autonomous underwater vehicle to effectively avoid all threat areas to find the optimal or suboptimal path with a small threat cost and fuel cost from the starting point to the target point.

Acknowledgments

This work was partially funded by the National Nature Science Foundation of China under grant No. 51679057, and partly supported by the Province Science Fund for Distinguished Young Scholars, China No. J2016JQ0052. The authors would like to thank the editor and anonymous reviewers for the helpful comments and suggestions.

References

[1] B. Akay, D. Karaboga, Artificial bee colony algorithm for large-scale problems and engineering design optimization, J. Intell. Manuf. 23 (4) (2012) 1001–1014.
[2] M. Azizi, R.G. Ejlali, S.A.M. Ghasemi, S. Talatahari, Upgraded whale optimization algorithm for fuzzy logic based vibration control of nonlinear steel structure, Eng. Struct. 192 (2019) 53–70.
[3] P. Barthelemy, J. Bertolotti, D.S. Wiersma, A, Lévy flight for light, Nature 453 (7194) (2008) 495–498.
[4] A. Baykasoğlu, Ş. Akpinar, Weighted superposition attraction (WSA): a swarm intelligence algorithm for optimization problems–part 2: constrained optimization, Appl. Soft. Comput. 37 (2015) 396–415.
[5] A. Baykasoğlu, F.B. Ozsoydan, Adaptive firefly algorithm with chaos for mechanical design optimization problems, Appl. Soft. Comput. 36 (2015) 152–164.
[6] Y. Cao, Y. Li, G. Zhang, K. Jermsittiparsert, M. Nasseri, An efficient terminal voltage control for PEMFC based on an improved version of whale optimization algorithm, Energy Rep. 6 (2020) 530–542.
[7] H. Chen, W. Li, X. Yang, A whale optimization algorithm with chaos mechanism based on quasi-opposition for global optimization problems, Expert Syst. Appl. 158 (2020) 113612.
[8] C.A.C. Coello, Constraint-handling using an evolutionary multiobjective optimization technique, Civ. Eng. Syst. 17 (4) (2000) 319–346.
[9] C.A.C. Coello, Use of a self-adaptive penalty approach for engineering optimization problems, Comput. Ind. 41 (2) (2000) 113–127.
[10] C.A.C. Coello, G.T. Pulido, M.S. Lechuga, Handling multiple objectives with particle swarm optimization, IEEE Trans. Evol. Comput. 8 (3) (2004) 256–279.

[11] K. Deb, Optimal design of a welded beam via genetic algorithms, AIAA J. 29 (11) (1991) 2013–2015.
[12] K. Deb, An efficient constraint handling method for genetic algorithms, Comput. Methods Appl. Mech. Engrg. 186 (2–4) (2000) 311–338.
[13] W. Du, M. Zhang, W. Ying, M. Perc, K. Tang, X. Cao, D. Wu, The networked evolutionary algorithm: A network science perspective, Appl. Math. Comput. 338 (2018) 33–43.
[14] H. Ebrahimgol, M. Aghaie, A. Zolfaghari, A. Naserbegi, A novel approach in exergy optimization of a WWER1000 nuclear power plant using whale optimization algorithm, Ann. Nucl. Energy 145 (2020) 107540.
[15] H. Eskandar, A. Sadollah, A. Bahreininejad, M. Hamdi, Water cycle algorithm–a novel metaheuristic optimization method for solving constrained engineering optimization problems, Comput. Struct. 110 (2012) 151–166.
[16] I. Fister, A. Iglesias, A. Galvez, J. Del Ser, E. Osaba, I. Fister Jr, M. Perc, M. Slavinec, Novelty search for global optimization, Appl. Math. Comput. 347 (2019) 865–881.
[17] I. Fister Jr, M. Perc, S.M. Kamal, I. Fister, A review of chaos-based firefly algorithms: perspectives and research challenges, Appl. Math. Comput. 252 (2015) 155–165.
[18] A.H. Gandomi, X.S. Yang, A.H. Alavi, Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems, Eng. Comput. 29 (1) (2013) 17–35.
[19] H. Garg, A hybrid PSO-GA algorithm for constrained optimization problems, Appl. Math. Comput. 274 (2016) 292–305.
[20] W. Gong, Z. Cai, Differential evolution with ranking-based mutation operators, IEEE Trans. Cybern. 43 (6) (2013) 2066–2081.
[21] Q. He, L. Wang, An effective co-evolutionary particle swarm optimization for constrained engineering design problems, Eng. Appl. Artif. Intell. 20 (1) (2007) 89–99.
[22] A.A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafaja, H. Chen, Harris hawks optimization: Algorithm and applications, Future Gener. Comput. Syst. 97 (2019) 849–872.
[23] G. Hou, L. Gong, Z. Yang, J. Zhang, Multi-objective economic model predictive control for gas turbine system based on quantum simultaneous whale optimization algorithm, Energy Convers. Manag. 207 (2020) 112498.
[24] T.J. Hsieh, A bacterial gene recombination algorithm for solving constrained optimization problems, Appl. Math. Comput. 231 (2014) 187–204.
[25] F. Huang, L. Wang, Q. He, An effective co-evolutionary differential evolution for constrained optimization, Appl. Math. Comput. 186 (1) (2007) 340–356.
[26] R. Jiang, M. Yang, S. Wang, T. Chao, An improved whale optimization algorithm with armed force program and strategic adjustment, Appl. Math. Model. 81 (2020) 603–623.
[27] S.S. Kannan, N. Ramaraj, A novel hybrid feature selection via symmetrical uncertainty ranking based local memetic search algorithm, Knowl.-Based Syst. 23 (6) (2010) 580–585.
[28] D. Karaboga, B. Basturk, A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm, J. Global Optim. 39 (3) (2007) 459–471.
[29] A. Kaveh, M. Khayatazad, A new meta-heuristic method: ray optimization, Comput. Struct. 112 (2012) 283–294.
[30] A. Kaveh, S. Talatahari, An improved ant colony optimization for constrained engineering design problems, Eng. Comput. 27 (1) (2010) 155–182.
[31] A. Kaveh, S. Talatahari, An improved ant colony optimization for constrained engineering design problems, Eng. Comput. 27 (1) (2010) 155–182.
[32] K.S. Lee, Z.W. Geem, A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice, Comput. Methods Appl. Mech. Engrg. 194 (36–38) (2005) 3902–3933.
[33] Z. Li, Y. Zhou, S. Zhang, J. Song, Lévy-flight moth-flame algorithm for function optimization and engineering design problems, Math. Probl. Eng. (2016) 1–22.
[34] Y. Ling, Y. Zhou, Q. Luo, Lévy flight trajectory-based whale optimization algorithm for global optimization, IEEE Access 5 (2017) 6168–6186.
[35] H. Liu, Z. Cai, Y. Wang, Hybridizing particle swarm optimization with differential evolution for constrained numerical and engineering optimization, Appl. Soft. Comput. 10 (2) (2010) 629–640.
[36] W. Long, W. Zhang, Y. Huang, Y. Chen, A hybrid cuckoo search algorithm with feasibility-based rule for constrained structural optimization, J. Cent. South Univ. 21 (8) (2014) 3197–3204.
[37] Y. Lu, Y. Zhou, X. Wu, A hybrid lightning search algorithm-simplex method for global optimization, Discrete Dyn. Nat. Soc. 2017 (2017) 1–23.
[38] Q. Luo, Y. Ling, Y. Zhou, Modified whale optimization algorithm for infinitive impulse response system identification, Arab. J. Sci. Eng. 45 (3) (2020) 2163–2176.
[39] Q. Luo, X. Yang, Y. Zhou, Nature-inspired approach: An enhanced moth swarm algorithm for global optimization, Math. Comput. Simulation 159 (2019) 57–92.
[40] M. Mahdavi, M. Fesanghary, E. Damangir, An improved harmony search algorithm for solving optimization problems, Appl. Math. Comput. 188 (2) (2007) 1567–1579.
[41] R. Menzel, U. Greggers, M. Hammer, Functional organization of appetitive learning and memory in a generalist pollinator, the honey bee, in: Insect Learning: Ecological and Evolutionary Perspectives, 1993, pp. 79–125.
[42] E. Mezura-Montes, C.A.C. Coello, An empirical study about the usefulness of evolution strategies to solve constrained optimization problems, Int. J. Gen. Syst. 37 (4) (2008) 443–473.
[43] E. Mezura-Montes, C.A.C. Coello, J. Velázquez-Reyes, L. Munozdavila, Multiple trial vectors in differential evolution for engineering design, Eng. Optim. 39 (5) (2007) 567–589.

[44] E. Mezura-Montes, J. Velázquez-Reyes, C.A.C. Coello, Modified differential evolution for constrained optimization, in: 2006 IEEE International Conference on Evolutionary Computation, IEEE, 2006, pp. 25–32.
[45] F. Miao, Y. Zhou, Q. Luo, Complex-valued encoding symbiotic organisms search algorithm for global optimization, Knowl. Inf. Syst. 58 (1) (2019) 209–248.
[46] S. Mirjalili, Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm, Knowl.-Based Syst. 89 (2015) 228–249.
[47] S. Mirjalili, Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems, Neural Comput. Appl. 27 (4) (2016) 1053–1073.
[48] S. Mirjalili, S.Z.M. Hashim, A new hybrid PSOGSA algorithm for function optimization, in: International Conference on Computer and Information Application, IEEE, 2010, pp. 374–377.
[49] S. Mirjalili, A. Lewis, The whale optimization algorithm, Adv. Eng. Softw. 95 (2016) 51–67.
[50] S. Mirjalili, S.M. Mirjalili, A. Hatamlou, Multi-verse optimizer: a nature-inspired algorithm for global optimization, Neural Comput. Appl. 27 (2) (2016) 495–513.
[51] S. Mirjalili, S.M. Mirjalili, A. Lewis, Grey wolf optimizer, Adv. Eng. Softw. 69 (2014) 46–61.
[52] A.A.A. Mohamed, A.A.M. El-Gaafary, Y.S. Mohamed, A.M. Hemeida, Multi-objective states of matter search algorithm for TCSC-based smart controller design, Electr. Power Syst. Res. 140 (2016) 874–885.
[53] B. Mohammadi, S. Mehdizadeh, Modeling daily reference evapotranspiration via a novel approach based on support vector regression coupled with whale optimization algorithm, Agric. Water Manage. 237 (2020) 106145.
[54] M. Montemurro, A. Vincenti, P. Vannucci, The automatic dynamic penalisation method (ADP) for handling constraints with genetic algorithms, Comput. Methods Appl. Mech. Engrg. 256 (2013) 70–87.
[55] V. Punnathanam, P. Kotecha, Yin-yang-pair optimization: A novel lightweight optimization algorithm, Eng. Appl. Artif. Intell. 54 (2016) 62–79.
[56] M.H. Qais, H.M. Hasanien, S. Alghuwainem, Enhanced whale optimization algorithm for maximum power point tracking of variable-speed wind generators, Appl. Soft. Comput. 86 (2020) 105937.
[57] K.M. Ragsdell, D.T. Phillips, Optimal design of a class of welded structures using geometric programming, J. Manuf. Sci. Eng. 98 (3) (1976) 1021–1025.
[58] R.V. Rao, V.J. Savsani, D.P. Vakharia, Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems, Comput.-Aided Des. 43 (3) (2011) 303–315.
[59] E. Rashedi, H. Nezamabadi-Pour, S. Saryazdi, GSA: a gravitational search algorithm, Inform. Sci. 179 (13) (2009) 2232–2248.
[60] T. Ray, K.M. Liew, Society and civilization: An optimization algorithm based on the simulation of social behavior, IEEE Trans. Evol. Comput. 7 (4) (2003) 386–396.
[61] A. Sadollah, A. Bahreininejad, H. Eskandar, M. Hamdi, Mine blast algorithm: A new population based algorithm for solving constrained engineering optimization problems, Appl. Soft. Comput. 13 (5) (2013) 2592–2612.
[62] A. Sadollah, H. Eskandar, J.H. Kim, Water cycle algorithm for solving constrained multi-objective optimization problems, Appl. Soft. Comput. 27 (2015) 279–298.
[63] P. Savsani, V. Savsani, Passing vehicle search (PVS): A novel metaheuristic algorithm, Appl. Math. Model. 40 (5–6) (2016) 3951–3978.
[64] Y. Sun, T. Yang, Z. Liu, A whale optimization algorithm based on quadratic interpolation for high-dimensional global optimization problems, Appl. Soft. Comput. 85 (2019) 105744.
[65] Y. Wang, Z. Cai, Y. Zhou, Z. Fan, Constrained optimization based on hybrid evolutionary algorithm and adaptive constraint-handling technique, Struct. Multidiscip. Optim. 37 (4) (2009) 395–413.
[66] Z. Wang, Q. Luo, Y. Zhou, Hybrid metaheuristic algorithm using butterfly and flower pollination base on mutualism mechanism for global optimization problems, Eng. Comput. (2020).
[67] F. Wilcoxon, Individual comparisons by ranking methods, Biom. Bull. 1 (6) (1945) 80–83.
[68] L.A. Wong, V.K. Ramachandaramurthy, S.L. Walker, P. Taylor, M.J. Sanjari, Optimal placement and sizing of battery energy storage system for losses reduction using whale optimization algorithm, J. Energy Storage 26 (2019) 100892.
[69] C. Yan, M. Li, W. Liu, Prediction of bank telephone marketing results based on improved whale algorithms optimizing s_Kohonen network, Appl. Soft. Comput. 92 (2020) 106259.
[70] X.S. Yang, A new metaheuristic bat-inspired algorithm, Comput. Knowl. Technol. 284 (2010) 65–74.
[71] J.H. Yi, L.N. Xing, G.G. Wang, J.Y. Dong, A.V. Vasilakos, A.H. Alavi, L. Wang, Behavior of crossover operators in NSGA-III for large-scale optimization problems, Inform. Sci. 509 (2020) 470–487.
[72] E. Zahara, Y.T. Kao, Hybrid nelder–mead simplex search and particle swarm optimization for constrained engineering design problems, Expert Syst. Appl. 36 (2) (2009) 3880–3886.
[73] Y. Zhang, Y. Liu, J. Li, J. Zhu, C. Yang, W. Yang, C. Wen, WOCDA: A whale optimization based community detection algorithm, Physica A 539 (2020) 122937.
[74] Q. Zhang, W. Liu, X. Meng, B. Yang, A.V. Vasilakos, Vector coevolving particle swarm optimization algorithm, Inform. Sci. 394 (2017) 273–298.
[75] H. Zhang, L. Tang, C. Yang, S. Lan, Locating electric vehicle charging stations with service capacity using the improved whale optimization algorithm, Adv. Eng. Inf. 41 (2019) 100901.
[76] J. Zhang, Y. Zhou, Q. Luo, Nature-inspired approach: a wind-driven water wave optimization algorithm, Appl. Intell. 49 (1) (2019) 233–252.
[77] Y.J. Zheng, Water wave optimization: a new nature-inspired metaheuristic, Comput. Oper. Res. 55 (2015) 1–11.
[78] Y. Zhou, Y. Ling, Q. Luo, Lévy flight trajectory-based whale optimization algorithm for engineering optimization, Eng. Comput. 35 (7) (2018) 2406–2428.
[79] Y. Zhou, S. Zhang, Q. Luo, M. Abdel-Baset, CCEO: cultural cognitive evolution optimization algorithm, Soft Comput. 23 (23) (2019) 12561–12583.