The Journal of Supercomputing (2023) 79:8789–8842
https://doi.org/10.1007/s11227-022-04996-1

ISSWOA: hybrid algorithm for function optimization and engineering problems

Jianhui Zhang[1] · Xuezhen Cheng[1] · Meng Zhao[1] · Jiming Li[1]

Accepted: 4 December 2022 / Published online: 26 December 2022

Abstract
A hybrid algorithm based on the sparrow search algorithm (SSA) and whale optimization algorithm (WOA) is proposed to address numerical and engineering optimization problems. The hybrid algorithm has enhanced global search ability through the WOA's improved spiral update mechanism, so that it does not easily fall into the local optimum. Further, using the guard mechanism of SSA introduced by the Levy flight, it has a strong ability to escape from the local optimum. The performance of the improved sparrow search whale optimization algorithm (ISSWOA) was investigated using 23 benchmark functions (classified into standard unimodal, multimodal, and fixed-dimension multimodal benchmark functions) and compared with similar algorithms. The experimental results indicated that ISSWOA was significantly superior to other algorithms on most benchmark functions. To evaluate the performance of ISSWOA in complex engineering problems, seven engineering design problems and a large electrical engineering problem were solved using ISSWOA. Compared with other algorithms, the results showed that ISSWOA had high potential for practical engineering problems.

Keywords Whale optimization · Sparrow search · Global optimization · Levy flight · Practical engineering designs

1 Introduction

Optimization problems are common in real-world applications, especially in physics, chemistry, and biology [1]. Typically, a fast and effective method is required to find the optimal solution to an optimization problem. The most effective of these is based on metaheuristics.

✉ Jiming Li
wugetongxue@163.com

1 College of Electrical Engineering and Automation, Shandong University of Science and Technology, Qingdao, China

Various metaheuristic algorithms exist. These can be classified into evolution-based algorithms (which imitate the law of the survival of the fittest), swarm-based algorithms (which simulate the cooperation and communication between animals), human-based algorithms (which imitate various human social behaviors), and physics-based algorithms (which imitate physical or chemical laws). An example of each type of algorithm is shown in Table 1.

Each metaheuristic algorithm has unique characteristics, and no algorithm outperforms others on all criteria. Different algorithms can be combined into a hybrid algorithm to leverage their characteristics, overcoming individual shortcomings and improving performance. Hybrid algorithms have been used in many studies. For example, Kumar et al. [22] proposed a multi-objective hybrid heat transfer search and passing vehicle search optimizer (MOHHTS–PVS) in which heat transfer search (HTS) acts as the main engine and passing vehicle search (PVS) is added as an auxiliary stage to enhance performance when applied to large engineering design problems. Yildiz et al. [23] proposed hybrid taguchi salp swarm algorithm-Nelder–Mead (HTSSA-NM) and hybrid artificial hummingbird algorithm and simulated annealing (HAHA-SA). HTSSA-NM used Nelder–Mead (NM) to improve the local search ability of the hybrid taguchi salp swarm algorithm (HTSSA), to optimize the structure and shape of an automobile brake pedal. HAHA-SA used simulated annealing (SA) to improve the performance of the artificial hummingbird algorithm (AHA), to solve constrained mechanical engineering problems. Li et al. [24] proposed particle swarm optimization-simulated annealing (PSO-SA) to complete seismic inversion in anisotropic media. PSO-SA combined simulated annealing (SA) and particle swarm optimization (PSO) and adjusted temperature by SA to control the particle aggregation jump out of local optima to improve the local search ability of PSO. Mafarja et al. [25] proposed whale optimization algorithm and simulated annealing (WOASA) to solve the problem of feature selection. WOASA embedded simulated annealing (SA) in the whale optimization algorithm (WOA) to enhance exploitation by searching the most promising regions located by WOA. Laskar et al. [26] proposed hybrid whale-particle swarm optimization (HWPSO), which combined particle swarm optimization (PSO) and the whale optimization algorithm, to solve electronic design optimization problems. HWPSO introduced “forced WOA” and “capping”; the former improves local optima avoidance in the exploration phase in PSO, and the latter accelerates the convergence to the global optimum. Han et al. [27] proposed the moth search-fireworks algorithm (MSFWA), which combined the moth search algorithm (MS) and fireworks algorithm (FA) to solve engineering design problems. MSFWA introduced the explosion and mutation operators from FA into MS to strengthen the exploitation capability of MS. Shehab et al. [28] proposed the cuckoo search and bat algorithm (CSBA), which combined the cuckoo search algorithm (CSA) and bat algorithm (BA) to solve numerical optimization problems. CSBA exploited the advantages of BA to improve the convergence ability of CSA.

SSA and WOA are swarm-based metaheuristic algorithms that imitate the predatory behavior of sparrows and whales, respectively. SSA [29] includes a step to help the algorithm escape local optima, which many other algorithms do not have; however, the effect is not obvious, and its global search ability is weak. WOA [30] has a strong local search ability owing to its spiral update mechanism

Table 1 Summary of partial metaheuristic algorithms

Evolution-based	Swarm-based	Human-based	Physics-based
Genetic algorithm (GA) [2]	Particle swarm optimization (PSO) [7]	Tabu search (TS) [14]	Simulated annealing (SA) [18]
Differential evolution (DE) [3]	Gray wolf optimization (GWO) [8]	Seeker optimization algorithm (SOA) [15]	Black hole algorithm (BH) [19]
Evolutionary strategy (ES) [4]	Snake optimization (SO) [9]	Social-based algorithm (SBA) [16]	Water cycle algorithm (WCA) [20]
Genetic programming (GP) [5]	Whale optimization algorithm (WOA) [10]	Exchange market algorithm (EMA) [17]	Ray optimization algorithm (RO) [21]
Gene expression programming (GEP) [6]	Dingo optimization algorithm (DOA) [11]		
	Sparrow search algorithm (SSA) [11]		
	African vultures optimization algorithm (AVOA) [13]		

but can easily fall into the local optimum because of its weak global search ability. The purpose of this research was to create a hybrid algorithm that could harness the strengths of each component to overcome these limitations. Accordingly, we improved WOA's spiral update mechanism to enhance its global search capability and used the Levy flight mechanism to improve SSA's ability to escape from the local optimum. Then, we combined the two improved algorithms into a hybrid algorithm that has strong global and local search capabilities and the ability to escape from the local optimum.

To evaluate the performance of ISSWOA, the hybrid algorithm was tested on the standard unimodal benchmark functions, standard multimodal benchmark functions, and standard fixed-dimensional multimodal benchmark functions. In addition, ISSWOA was applied to seven engineering design problems and an electrical engineering problem. The results showed that ISSWOA had strong optimization ability and was a competitive algorithm for solving practical problems.

The remainder of this paper is organized as follows. Sections 2 and 3 describe SSA and WOA, respectively. Section 4 details the proposed ISSWOA. The application of the proposed algorithm to 23 benchmark functions and engineering problems is reported in Sects. 5 and 6, respectively. Finally, Sect. 7 summarizes the paper and outlines future research directions.

2 SSA

SSA [31] is a metaheuristic optimization algorithm based on the sparrows' foraging and antipredation behaviors and describes a discoverer–follower model with an awareness mechanism. In SSA, sparrows with high fitness in the population are regarded as producers, while others are considered as scroungers, and a proportion of individuals in the population are selected as guards for detection of threats.

2.1 Producers

After initializing the sparrow population, all sparrows are ranked by their fitness, and some of the individuals with better fitness are selected as producers. The producers' iterative equation is given by

$$\overrightarrow{X_i}(t+1) = \begin{cases} \overrightarrow{X_i}(t) * \exp\left(\frac{-i}{\alpha \cdot Maxiter}\right), & R < ST \\ \overrightarrow{X_i}(t) + Q \cdot L, & R > ST \end{cases}, \tag{1}$$

where t is the current iteration, $\overrightarrow{X_i}(t)$ is the current location of the ith sparrow, *Maxiter* is the maximum number of iterations, α is a random number in (0,1], R is a random number in [0,1], ST in [0.5,1] is a safety threshold, Q is a random number drawn from a normal distribution, and L is a $1 \times D$ matrix of ones.

2.2 Scrounger

After the producers update their positions, the other individuals are selected as scroungers, and their position update equation is given by

$$\overrightarrow{X_i}(t+1)=\begin{cases} Q\cdot\exp\left(\frac{\overrightarrow{X_{\text{worst}}}(t)-\overrightarrow{X_i}(t)}{i^2}\right), & i>\frac{n}{2} \\ \overrightarrow{X_{\text{P}}}(t)+\left|\overrightarrow{X_i}(t)-\overrightarrow{X_P}(t)\right|\cdot A^{+}\cdot L, & i\le\frac{n}{2} \end{cases}, \tag{2}$$

where $\overrightarrow{X_{\text{worst}}}(t)$ is the worst position globally, n is the number of sparrows, $\overrightarrow{X_{\text{P}}}(t)$ is the best position of producers, and A is a $1\times D$ matrix with each element randomly assigned the value of 1 or -1.

2.3 Guard

After the positions of producers and scroungers are updated, a proportion of sparrows are selected from the population as guards, and their update equation is given by

$$\overrightarrow{X_i}(t+1)=\begin{cases} \overrightarrow{X_{\text{best}}}(t)+\beta*\left|\overrightarrow{X_i}(t)-\overrightarrow{X_{\text{best}}}(t)\right|, & f_i\neq f_g \\ \overrightarrow{X_i}(t)+K\cdot\left(\frac{\left|\overrightarrow{X_i}(t)-\overrightarrow{X_{\text{worst}}}(t)\right|}{(f_i-f_{\text{w}})+\varepsilon}\right), & f_i=f_g \end{cases}, \tag{3}$$

where $\overrightarrow{X_{\text{best}}}(t)$ is the current optimal position, β is a random number that controls the step size and obeys a Gaussian distribution, f_{i} is the fitness of the ith sparrow, f_{g} is the optimal fitness, K is a random number in $(-1,1)$, and f_{w} is the worst fitness.

The pseudocode and flowchart of SSA are shown in Figs. 1 and 2, respectively.

3 WOA

WOA [32] is a metaheuristic optimization algorithm that mimics the hunting behavior of humpback whales. Humpback whales hunt using a bubble-net strategy. In WOA, this strategy is mathematically modeled as detailed below.

3.1 Encircling prey

For encircling prey, the whale herd moves iteratively toward the current optimal position according to the following iterative equation:

$$\vec{D}=\left|\vec{C}\cdot\overrightarrow{X^*}(t)-\vec{X}(t)\right|, \tag{4}$$

Pseudocode of SSA:

```
Define:
Maxiter: maximum number of iterations
N: the number of salps
1. Initialize population of  salps
2. While t < Maxiter
3.    Rank the fitness and choose the first N/2 salps with better fitness values as leaders
4.    The rest of salps are followers
4.    For i = 1: D
5.        Update position of current producer using equation (1)
6.    End for
7.    For i =  D: N
8.        Update position of current scrounger using equation (2)
9.    End for
10.   For j =  1: S
11.       Obtain a new position using equation (3)
12.       If new position is better than the previous position, update it
13.   End for
14. End while
15. Return best solution.
```

Fig. 1 Pseudocode of SSA

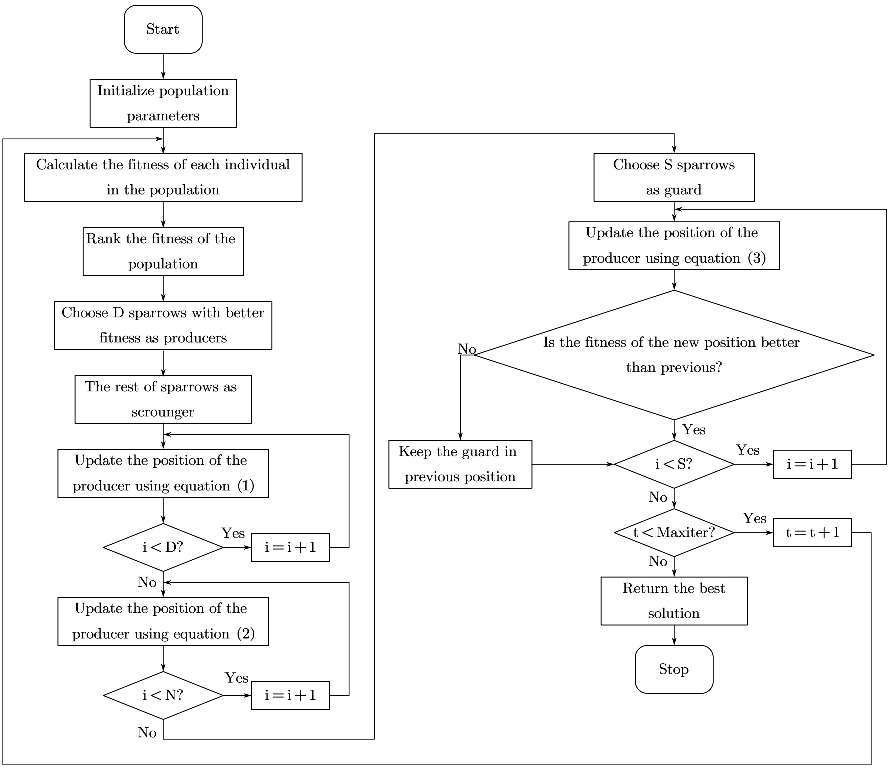

Fig. 2 Flowchart of SSA

$$\vec{X}(t+1) = \overrightarrow{X^*}(t) - \vec{A} \cdot \vec{D}, \tag{5}$$

where $\vec{A}$ and $\vec{C}$ are vector coefficients, $\overrightarrow{X^*}(t)$ is the current optimal position, $\vec{X}(t)$ is the current position of the whale, t is the current iteration, and $\vec{A}$ and $\vec{C}$ are expressed as

$$\vec{A} = 2\vec{a} \cdot \vec{r} - \vec{a}, \tag{6}$$

$$\vec{C} = 2 \cdot \vec{r}, \tag{7}$$

where $\vec{r}$ is a random vector with values in [0,1] and $\vec{a}$ decreases from 2 to 0 over the iterations as follows:

$$\vec{a} = 2 - \frac{2t}{MaxIter}, \tag{8}$$

with *Maxiter* being the maximum number of iterations.

3.2 Bubble-net hunting

The bubble-net strategy of whales can adopt one out of two behaviors: encircling the prey and spiral updating position.

Encircling the prey is achieved by decreasing $\vec{a}$ linearly from 2 to 0 over the iterative process to gradually reduce the fluctuation range of $\vec{A}$. Vector $\vec{r}$ takes a random value in [0,1] for $\vec{A}$ to take a random number in $[-a, a]$. Hence, whales can appear anywhere at random between their current position and the current best position. For spiral updating position, a spiral equation between the position of the current whale and prey is created to model the following helix-shaped movement of whales:

$$\vec{X}(t+1) = \overrightarrow{D'} \cdot e^{bl} \cdot \cos(2\pi l) + \overrightarrow{X^*}(t), \tag{9}$$

$$\overrightarrow{D'} = \left| \overrightarrow{X^*}(t) - \vec{X}(t) \right|, \tag{10}$$

where b is a constant with value 1 to define the helix shape and l is a random number in [− 1,1].

Whales randomly choose one hunting behavior with equal probability as follows:

$$\vec{X}(t+1) = \begin{cases} \overrightarrow{X^*}(t) - \vec{A} \cdot \vec{D} & p < 0.5 \\ \overrightarrow{D'} \cdot e^{bl} \cdot \cos(2\pi l) + \overrightarrow{X^*}(t) & p > 0.5 \end{cases}, \tag{11}$$

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

3.3 Search for prey

To localize a prey, whales move to the random positions rather than to the optimal position. The condition for whales to perform this behavior is $\left|\vec{A}\right| > 1$, and the behavior is described by

$$\overrightarrow{D'} = \left|\vec{C} \cdot \overrightarrow{X_{\text{rand}}}(t) - \vec{X}(t)\right|, \tag{12}$$

$$\vec{X}(t+1) = \overrightarrow{X_{\text{rand}}}(t) - \vec{A} \cdot \overrightarrow{D'}, \tag{13}$$

where $\overrightarrow{X_{\text{rand}}}(t)$ is the random position of a whale.

The pseudocode and flowchart of WOA are shown in Figs. 3 and 4, respectively.

4 Proposed ISSWOA

4.1 Improved SSA

Levy flight [33] uses Levy random number to control the individual step size for the individuals to move randomly. As shown in Fig. 5, using K which control the pace of the guard in the original SSA and Levy generate 100 random numbers, different from the random numbers generated by K, the random numbers generated by Levy are not all in [− 1,1], a small part exceed this range, using Levy random number to control individuals step size and they can move in small steps

Pseudocode of WOA

```
Define
Maxiter: maximum number of iterations
p : the number of whales
1. Initialize population of whales
2. While p< Maxiter
3.     Calculate the fitness of each whale in the population.
4.     For p = 1:p
5.         Update p,p,p,pand p
6.         If p < 0.5
7.             If |p| <1
8.                 Update position of current whale using equation (5)
9.             Else
10.                Update position of current whale using equation (13)
11.            End if
12.        Else
13.            Update position of current whale using equation (9)
14.        End if
15.    End for
16. End while
17. Return best solution.
```

Fig. 3 Pseudocode of WOA

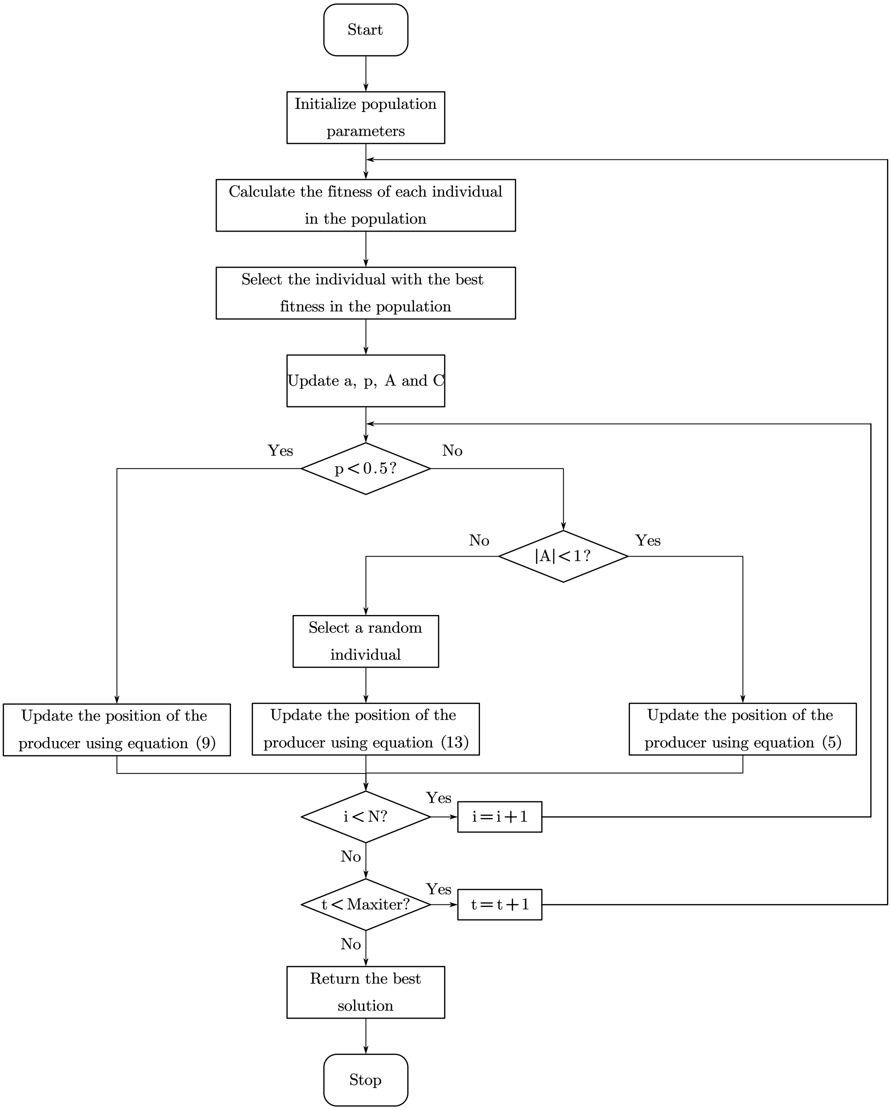

Fig. 4 Flowchart of WOA

with a large probability and large steps with a small probability; in this way, we can strengthen the ability of SSA to escape from the local optimum while ensuring the local search ability of the algorithm. Thus, Eq. (3) is rewritten as

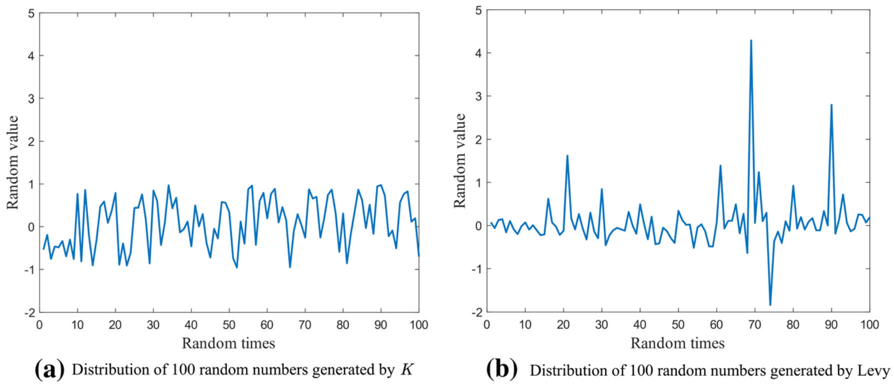

Fig. 5 Distribution of 100 random numbers generated by K and Levy

$$\overrightarrow{X_i}(t+1) = \begin{cases} \overrightarrow{X_{\text{best}}}(t) + \text{Levy} \cdot \left| \overrightarrow{X_i}(t) - \overrightarrow{X_{\text{best}}}(t) \right|, & f_i \neq f_g \\ \overrightarrow{X_i}(t) + \text{Levy} \cdot \left(\frac{\left| \overrightarrow{X_i}(t) - \overrightarrow{X_{\text{worst}}}(t) \right|}{(f_i - f_w) + \varepsilon} \right), & f_i = f_g \end{cases}, \tag{14}$$

$$\text{Levy} = 0.1 \cdot \frac{u}{|v|^{\frac{1}{\beta}}}, \tag{14a}$$

$$u \sim N\left(0, \sigma_u^2\right), \quad \sigma_u^2 = \left(\frac{\Gamma(1+\beta) \sin\left(\frac{\beta\pi}{2}\right)}{\Gamma\left(\frac{1+\beta}{2}\right) 2^{(\beta-1)/2} \beta} \right)^{\frac{1}{\beta}}, \tag{14b}$$

$$v \sim N\left(0, \sigma_v^2\right), \quad \sigma_v^2 = 1, \tag{14c}$$

where β is 1.5, and u and v are random numbers drawn from normal distributions with zero mean and variances σ_u^2 and σ_v^2, respectively.

4.2 Improved WOA

Spiral updating in WOA is a search method centered at the current optimal position. When the whales are concentrated at a later optimization stage, spiral updating improves the local search ability of WOA. However, global search capability of WOA is weak. To prevent this problem, we define the following spiral updating equation:

$$\overrightarrow{X_i}(t+1) = \overrightarrow{X_i}(t) + \alpha(i) \cdot \left(\overrightarrow{X_i}(t) - \overrightarrow{X_{\text{mean}}}(t) \right) + \beta(i) \cdot \left(\overrightarrow{X_i}(t) - \overrightarrow{X_{i+1}}(t) \right), \tag{15}$$

$$\alpha(i) = \frac{\rho(i)}{\max(|xr|)}, \quad \beta(i) = \frac{\tau(i)}{\max(|yr|)}, \tag{15a}$$

$$\rho(i) = \varphi(i) \cdot \sin(\omega(i)), \quad \tau(i) = \varphi(i) \cdot \cos(\omega(i)), \tag{15b}$$

$$\varphi(i) = \omega(i) + R \cdot \text{rand}, \quad \omega(i) = c \cdot \pi \cdot \text{rand}, \tag{15c}$$

where $\overrightarrow{X_{\text{mean}}}(t)$ is the average position of the current population, $\overrightarrow{X_{i+1}}(t)$ is the position of the next whale to be updated, R is a random number in [0.5,2], rand is a random number in [0,1], and c is a random number in [5, 10].

The new spiral updating [34] is centered at the current whale and carries out a spiral search outward to determine the existence of a better surrounding position. This update is adopted before the algorithm completes two-thirds of all the iterations. Then, Eq. (9) is adopted. Hence, the convergence accuracy is ensured by updating the position following Eq. (9) at a later stage, and the global search ability is improved by updating the position according to Eq. (15).

4.3 ISSWOA

The proposed ISSWOA combines our improved versions of WOA and SSA. Combining the two algorithms allows to fully use their advantages while overcoming the shortcomings of the original algorithms, thereby ensuring that the optimal solution can be found faster and more effectively.

The pseudocode and flowchart of the proposed ISSWOA are shown in Figs. 6 and 7, respectively. ISSWOA first gives each individual a random position within the search space to create an initial population. Then, it orders the initial population by fitness and chooses a proportion of the individuals with better fitness as producers and the remaining individuals as scroungers. After the producer updates its position, the scrounger randomly chooses to update its position according to Eq. (2) or the whale's encircling behavior. After the producer's position is updated, the population uses spiral updating to find a better position near itself or the optimal solution. Finally, some individuals are selected as guards, and their positions are updated according to Eq. (14). The iterative process is repeated until the optimization criteria are met.

When the population updates its position by spiral updating, if the number of iterations of the population is two-thirds of the maximum number of iterations, the position is updated by Eq. (14). Otherwise, the position is updated by Eq. (9). If the new position is better than the previous one, the position is updated; otherwise, the previous position is maintained.

Spiral updating in WOA strengthens the global searching ability of the individuals. Combined with the alert mechanism of SSA, ISSWOA prevents easily falling into a local optimum and improves the convergence accuracy.

```
Pseudocode of ISSWOA:
Define:
Maxiter: maximum number of iterations
N: the number of individuals
D: the number of producers
S: the number of guard
Initialize population.
While t< Maxiter
    Rank the fitness and find the current best individual and the current worst individual
    For i = 1 : D
        Update position of current producer using equation (1)
    End for
    For i = D: N
        Update a, A, C, l and p
        If p < 0.5
            If |A| <1
                Update position of current scrounger using equation (5)
            Else
                Update position of current scrounger using equation (13)
            End if
        Else
                Update position of current scrounger using equation (2)
        End if
    End for
    For i = 1: N
        If t < 2/3 * Maxiter
            Obtain a new position using equation (15)
        Else
            Obtain a new position using equation (9)
        End if
        If new position is better than the previous position, update it
    End for
    For j = 1: S
        Obtain a new position using equation (14)
        If new position is better than the previous position, update it
    End for
End while
Return best solution.
```

Fig. 6 Pseudocode of the proposed ISSWOA

5 Performance evaluation of ISSWOA on benchmark functions

We evaluated results for 100 independent runs of ISSWOA on different benchmark functions and compared it with similar algorithms to verify its superiority.

5.1 Algorithms for comparison

In this research, we compared ISSWOA with the original WOA, SSA, improved SSA (ISSA), and improved WOA (IWOA) to verify the improvement in the algorithm. Then we compared ISSWOA with snake optimization (SO), the pelican optimization algorithm (POA), the grasshopper optimization algorithm (GOA), improved

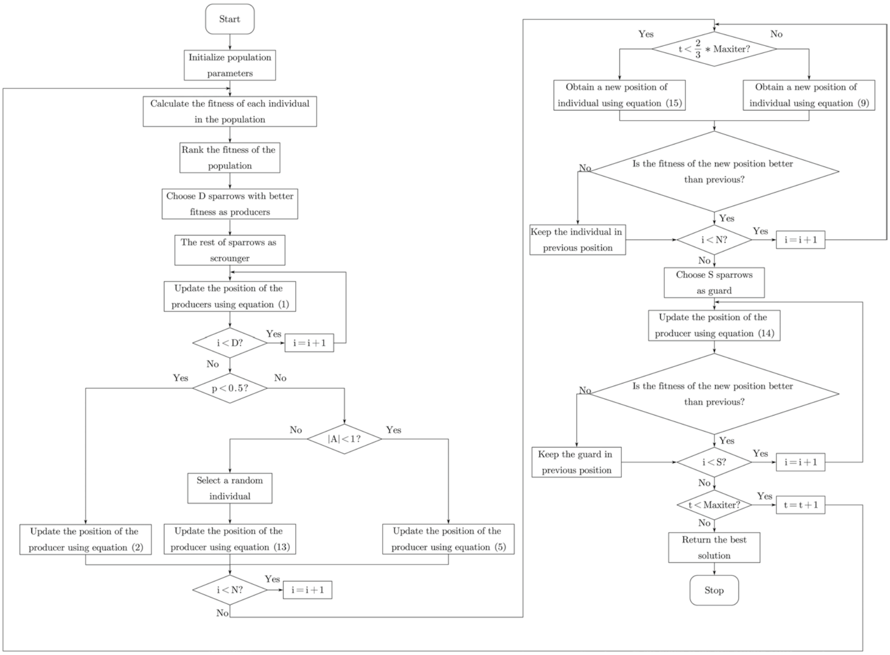

Fig. 7 Flowchart of the proposed ISSWOA

gray wolf optimization (I-GWO) [35], and the hybrid particle swarm optimization and butterfly optimization algorithm (HPSOBOA) [36] to verify the optimization capability. GOA, DOA, SO, and POA are relatively novel metaheuristic algorithms; I-GWO and HPSOBOA are improved algorithms for basic metaheuristic algorithms.

5.2 Parameter settings

The most obvious parameters that affected the performance of ISSWOA were the number of leaders and number of guards. Initially, 60 positions were randomly generated for the population, and Eq. (1) with $R < ST$ was used to iterate the population 100 times. From the results (Fig. 8), it can be seen that the position of the leader will approach the origin as the number of iterations increases. Consequently, if the number of leaders is set too high, and the optimal solution is far from the origin, the algorithm will have poor performance.

The number of guards is not the more the better. After testing, if there are too many guards, the ISSWOA test results of F6, F7, and F8, etc., will become worse. On the contrary, if there are too few guards, the results of ISSWOA for F12 and F13, etc., will also become worse.

From the analysis and early experimental testing, when the population number was set to 100, the number of leaders set at 40, and the number of guards set at 30, the performance of the algorithm was more balanced for most benchmark functions.

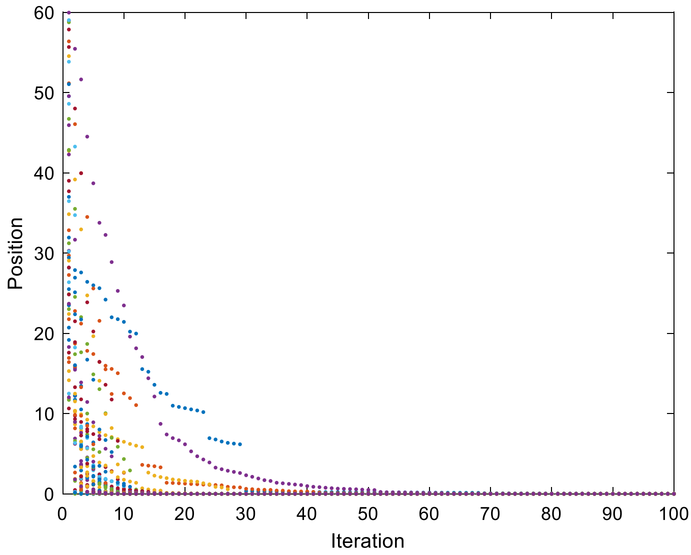

Fig. 8 Population iteration graph for discoverer when $R < ST$

The population number of other comparison algorithms was also set at 100, and the other parameter settings were the same as those in the original literature.

5.3 Benchmark functions

In this study, ISSWOA was applied to standard unimodal benchmark functions, multimodal benchmark functions, and fixed-dimensional multimodal benchmark functions [37] for testing and comparison with other algorithms. The functional formulas of benchmark functions are shown in Tables 2, 3, and 4, and their 3D views are shown in Figs. 9, 10, and 11.

Table 2 Standard unimodal benchmark functions

Function	Dimension	Range	f_{min}
$F_1(x) = \sum_{i=1}^{n} x_i^2$	30	[− 100,100]	0
$F_2(x) = \sum_{i=1}^{n} \lvert x_i \rvert = \prod_{i=1}^{n} \lvert x_i \rvert$	30	[− 10,10]	0
$F_3(x) = \sum_{i=1}^{n} \left(\sum_{j=1}^{i} x_j \right)^2$	30	[− 100,100]	0
$F_4(x) = \max_i \left\{ \lvert x_i \rvert, \quad 1 \le i \le n \right\}$	30	[− 100,100]	0
$F_5(x) = \sum_{i=1}^{n-1} \left[100 \left(x_{i+1} - x_i^2 \right)^2 + \left(x_i - 1 \right)^2 \right]$	30	[− 30,30]	0
$F_6(x) = \sum_{i=1}^{n} \left([x_i + 0.5] \right)^2$	30	[− 100,100]	0
$F_7(x) = \sum_{i=1}^{n} i x_i^4 + \text{random}[0, 1)$	30	[− 1.28,1.28]	0

Table 3 Standard multimodal benchmark functions

Function	Dimension	Range	$f_{\min}$
$F_8(x) = \sum_{i=1}^{n} -x_i \sin\left(\sqrt{\lvert x_i \rvert}\right)$	30	[−500,500]	−418.9829*D
$F_9(x) = \sum_{i=1}^{n} \left[x_i^2 - 10\cos\left(2\pi x_i\right) + 10\right]$	30	[−5.12,5.12]	0
$F_{10}(x) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos\left(2\pi x_i\right)\right) + 20 + e$	30	[−32,32]	8.8818E−16
$F_{11}(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right)$	30	[−600,600]	0
$F_{12}(x) = \frac{\pi}{n}\left\{10\sin\left(\pi y_1\right) + \sum_{i=1}^{n-1}\left(y_i - 1\right)^2\left[1 + 10\sin^2\left(\pi y_{i+1}\right)\right] + \left(y_n - 1\right)^2\right\} + \sum_{i=1}^{n} u\left(x_i, 10, 100, 4\right) y_i = 1 + \frac{x_i + 1}{4} u\left(x_i, a, k, m\right) = \begin{cases} k\left(x_i - a\right)^m x_i > a \\ 0 - a < x_i < a \\ k\left(-x_i - a\right)^m x_i < -a \end{cases}$	30	[−50,50]	0
$F_{13}(x) = 0.1\left\{\sin^2\left(3\pi x_1\right) + \sum_{i=1}^{n}(x_i - 1)^2\left[1 + \sin^2\left(3\pi x_i + 1\right)\right] + (x_n - 1)^2[1 + \sin^2(2\pi x_n)]\right\} + \sum_{i=1}^{n} u\left(x_i, 5, 100, 4\right)$	30	[−50,50]	0

5.4 Analysis of ISSWOA for standard unimodal benchmark functions

Table 5 shows the test results of the algorithm on the standard unimodal benchmark functions. It can be seen from the result that IWOA was better than WOA, especially in finding the optimal values of F1, F3, and F5. Because the improvement in the original WOA spiral mechanism significantly enhanced its global search ability, it could quickly find a good approximate position in the early stage and converge more fully by the end. For the ISSA, Levy flight mechanism mainly improved the exploration ability of the algorithm to help it escape local optima; it played only a limited role in the standard unimodal benchmark functions; therefore, the gap between ISSA and SSA was relatively small. Compared with the other algorithms, ISSWOA found the global optimal value for F1, F2, F3, and F4. The optimal values for F5, F6, and F7 found by ISSWOA were superior to those found by other algorithms; in particular, the optimal value for F5 was superior owing to the improved spiral update mechanism of IWOA and discoverer mechanism of ISSA. The above results showed that ISSWOA had good performance on standard unimodal benchmark functions.

Figures 12 and 13 display the iteration curves and box diagrams for ISSWOA and the other algorithms on standard unimodal benchmark functions, respectively. It can be seen from the figure that the iteration speed of ISSWOA was superior to that of other algorithms on seven standard unimodal benchmark functions, and its box diagram shows that its 100 times optimal results were distributed in a centralized manner, indicating that ISSWOA had good robustness.

Table 4 Standard fixed-dimensional multimodal benchmark functions

Function	Dimension	Range	$f_{\min}$
$F_{14}(x)=\left(\frac{1}{500}+\sum_{j=1}^{25}\frac{1}{j+\sum_{i=1}^{2}\left(x_i-a_{ij}\right)^6}\right)^{-1}$	2	[− 65.53,65.53]	0.998
$F_{15}(x)=\sum_{i=1}^{11}\left[a_i-\frac{x_i\left(b_i^2+b_ix_2\right)}{b_i^2+b_ix_3+x_4}\right]^2$	4	[− 5,5]	0.0003075
$F_{16}(x)=4x_1^2-2.1x_1^4+\frac{1}{3}x_1^6+x_1x_2-4x_2^2+4x_2^4$	2	[− 5,5]	− 103,163
$F_{17}(x)=\left(x_2-\frac{5.1}{4\pi^2}x_1^2+\frac{5}{\pi}x_1-6\right)^2+10\left(1-\frac{1}{8\pi}\right)\cos x_1+10$	2	[− 5,10] × [0,15]	0.398
$F_{18}(x)=\left[1+\left(x_1+x_2+1\right)^2\left(19-14x_1+3x_1^2-14x_2+6x_1x_2+3x_2^2\right)\right]\times\left[30+\left(2x_1-3x_2\right)^2\left(18-32x_1+12x_1^2+48x_2-36x_1x_2+27x_2^2\right)\right]$	2	[− 5,5]	3
$F_{19}(x)=-\sum_{i=1}^{4}c_i\exp\left(-\sum_{j=1}^{3}a_{ij}\left(x_j-p_{ij}\right)^2\right)$	3	[0,1]	− 3.8628
$F_{20}(x)=-\sum_{i=1}^{4}c_i\exp\left(-\sum_{j=1}^{6}a_{ij}\left(x_j-p_{ij}\right)^2\right)$	6	[0,1]	− 3.32
$F_{21}(x)=-\sum_{i=1}^{5}\left[\left(x-a_i\right)\left(x-a_i\right)^T+c_i\right]^{-1}$	4	[0,10]	− 10.1532
$F_{22}(x)=-\sum_{i=1}^{7}\left[\left(x-a_i\right)\left(x-a_i\right)^T+c_i\right]^{-1}$	4	[0,10]	− 10.4029
$F_{23}(x)=-\sum_{i=1}^{10}\left[\left(x-a_i\right)\left(x-a_i\right)^T+c_i\right]^{-1}$	4	[0,10]	− 10.5364

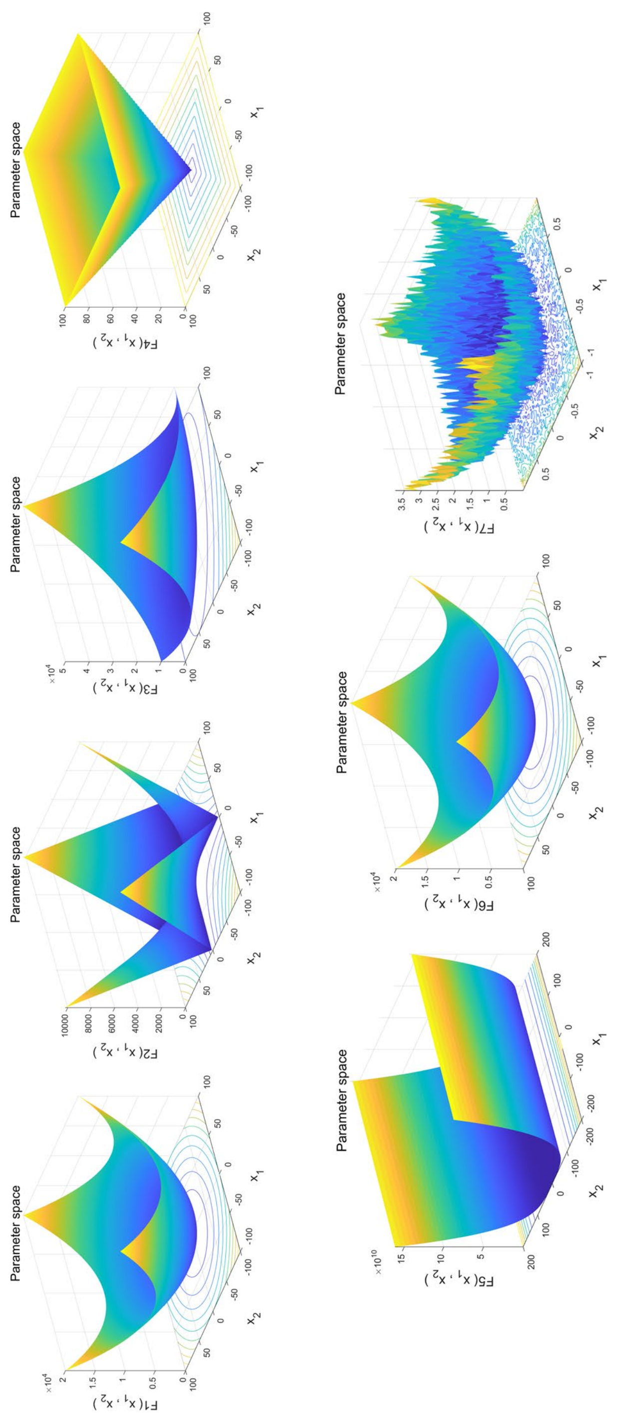

Fig. 9 3D view of standard unimodal benchmark functions

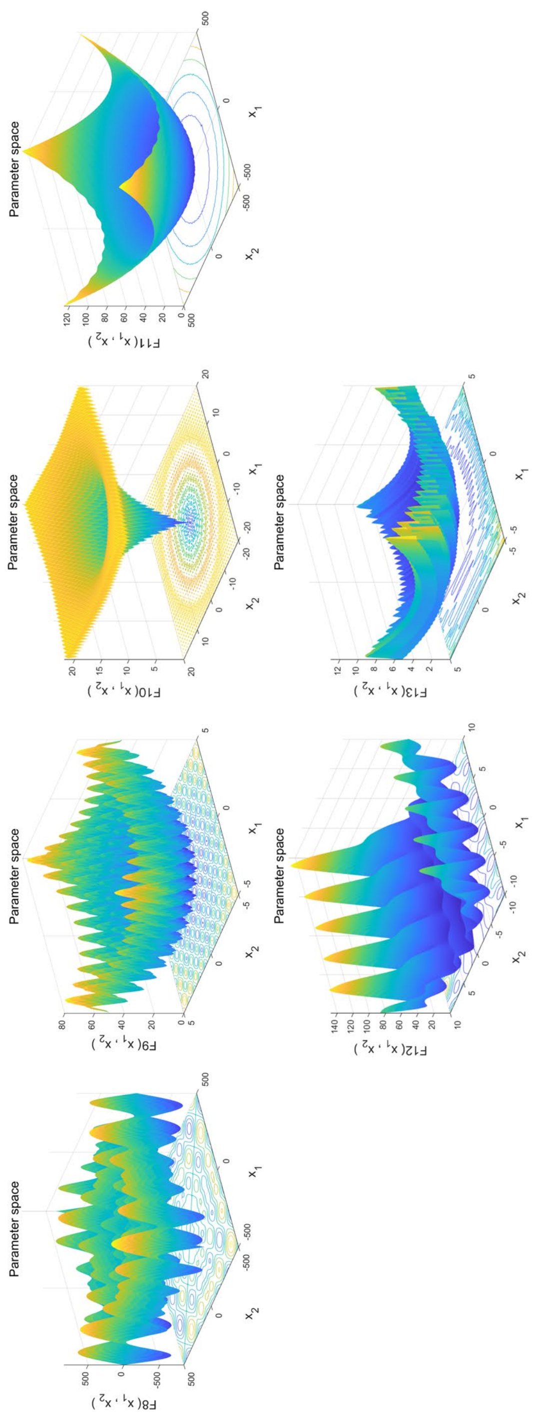

Fig. 10 3D view of standard multimodal benchmark functions

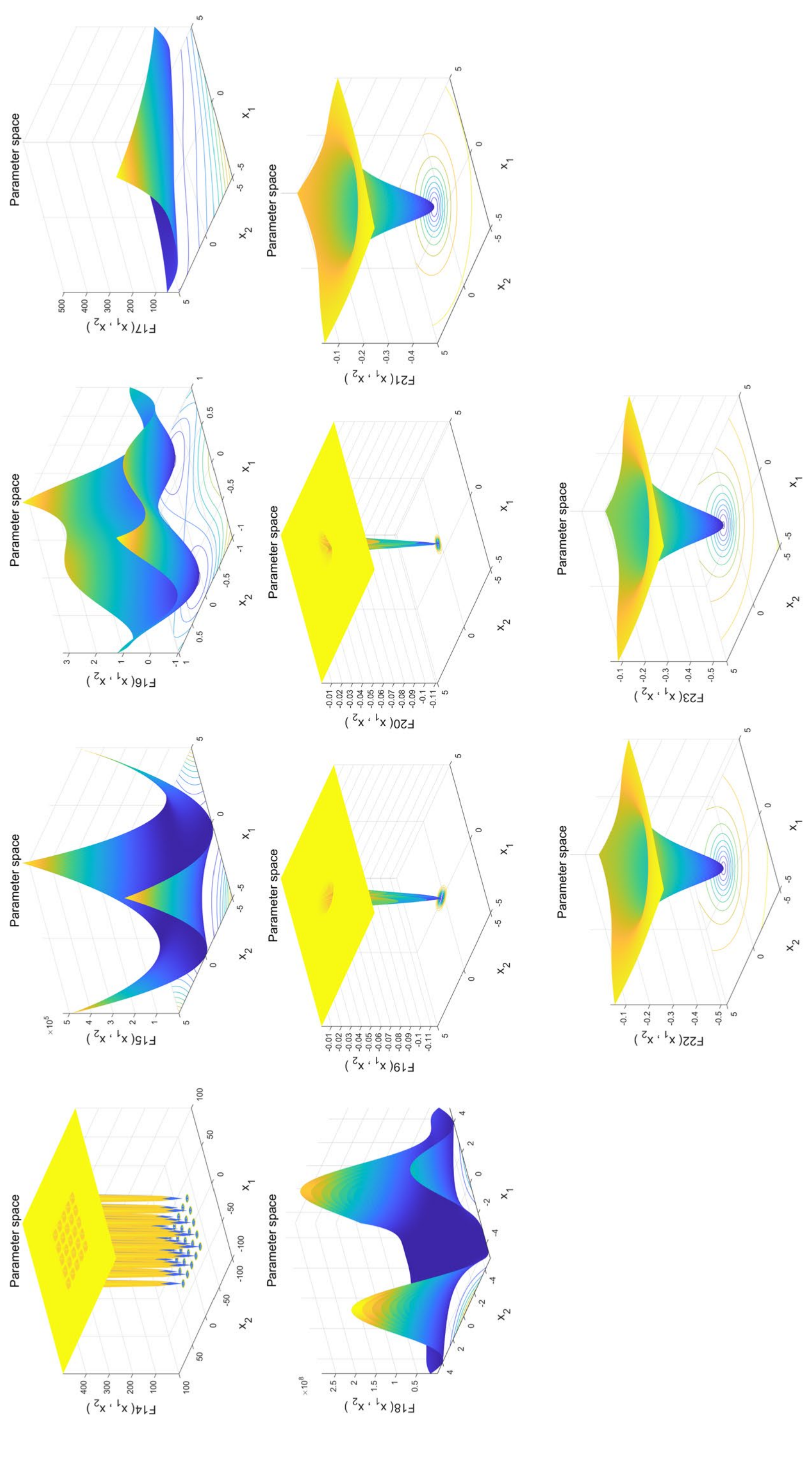

Fig. 11 3D view of standard fixed-dimensional multimodal benchmark functions

Table 5 Comparative result of ISSWOA with other algorithms for standard unimodal benchmark functions

Algorithms	Parameters	F1	F2	F3	F4	F5	F6	F7
SSA	Mean	2.6000E−45	1.5300E−38	2.8900E−62	9.4300E−41	2.8509E+01	7.4901E+00	7.8760E−04
	Std	2.6000E−44	1.5300E−37	2.8900E−61	9.4300E−40	1.0601E−01	9.3968E−02	5.8404E−04
	Best	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	2.8304E+01	6.5608E+00	3.3000E−05
	Worst	2.6000E−43	1.5300E−36	2.8900E−60	9.4300E−39	2.8705E+01	7.5000E+00	2.5620E−03
WOA	Mean	2.8216E−57	2.1994E−34	1.0392E+03	2.0061E−09	2.7364E+01	3.5608E−02	1.7409E−03
	Std	2.2749E−56	8.7666E−34	2.9570E+03	6.2162E−09	3.5498E−01	3.3711E−02	2.2362E−03
	Best	4.4050E−66	4.4204E−41	1.8200E−05	2.5900E−18	2.6409E+01	6.4290E−03	7.5827E−06
	Worst	2.2441E−55	6.0206E−33	1.9949E+04	4.8200E−08	2.8711E+01	2.8475E−01	1.1367E−02
ISSA	Mean	8.4106E−65	3.1229E−40	9.6973E−111	2.2165E−42	2.6013E+01	2.3695E−03	3.2813E−04
	Std	8.4106E−64	3.1229E−33	9.6972E−110	2.2165E−41	1.4777E−01	6.4190E−04	2.6896E−04
	Best	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	2.5578E+01	1.1331E−03	1.4272E−05
	Worst	8.4106E−63	3.1229E−32	9.6972E−109	2.2165E−40	2.6277E+01	4.6003E−03	1.3000E−03
IWOA	Mean	1.8501E−80	3.0705E−41	4.0520E−06	4.3979E−49	2.2161E−01	1.7526E−03	5.2870E−04
	Std	1.8500E−79	2.8717E−40	2.2701E−05	4.3698E−48	3.5249E−01	3.8022E−03	5.9091E−04
	Best	3.6500E−122	8.6100E−62	2.6200E−13	2.7700E−65	3.5400E−04	1.2600E−05	3.5300E−06
	Worst	1.8500E−78	2.8700E−39	2.1258E−04	4.3700E−47	2.0100E+00	3.1600E−02	3.2546E−03
SO	Mean	9.8258E−59	2.0856E−25	8.8001E−36	3.0815E−25	3.4805E+01	2.4924E−01	1.9045E−04
	Std	3.0840E−58	1.1018E−24	4.0869E−35	5.9778E−25	3.6519E+01	1.4747E−01	1.3456E−04
	Best	8.1478E−62	1.1026E−28	3.9737E−42	1.0072E−27	7.9926E−03	1.0377E−04	2.5833E−05
	Worst	2.3526E−57	1.0861E−23	3.0353E−34	3.4165E−24	1.6194E+02	7.6774E−01	6.7656E−04
POA	Mean	3.0596E−57	2.7168E−30	2.2051E−57	4.4898E−30	2.8255E+01	2.6452E+00	1.6382E−04
	Std	2.1066E−56	1.0748E−29	1.8093E−56	1.6839E−29	6.3250E−01	5.9797E−01	1.0949E−04
	Best	6.0163E−76	1.2893E−36	4.0258E−71	1.1465E−35	2.6372E+01	1.1659E+00	1.2504E−05
	Worst	1.8489E−55	7.3801E−29	1.7872E−55	1.1842E−28	2.8865E+01	4.3349E+00	5.0726E−04

Table 5 (continued)

Algorithms	Parameters	F1	F2	F3	F4	F5	F6	F7
DOA	Mean	1.0054E−125	2.6491E−66	7.7698E−128	9.4462E−62	2.8975E+01	6.7731E+00	4.8254E−04
	Std	5.3890E−125	1.2045E−65	7.0296E−127	5.8426E−61	1.9230E−02	5.1080E−01	4.7837E−04
	Best	3.7999E−139	9.4669E−73	3.7361E−142	8.3150E−67	2.8896E+01	4.8146E+00	1.5968E−06
	Worst	4.6317E−124	1.1390E−64	7.0063E−126	5.7427E−60	2.8998E+01	7.4556E+00	3.0911E−03
GOA	Mean	1.9106E−01	4.2078E+00	8.6451E+02	1.6135E+00	2.2697E+02	1.7376E−01	5.4736E−02
	Std	5.1217E−02	1.0695E+01	1.9355E+03	9.4687E−01	4.5467E+02	4.6754E−02	2.5327E−02
	Best	6.7388E−02	2.2038E−01	7.0825E+01	3.8003E−01	2.9489E+01	7.9446E−02	1.9862E−02
	Worst	3.3949E−01	8.8869E+01	1.2748E+04	4.5012E+00	3.1225E+03	2.8485E−01	1.4205E−01
I-GWO	Mean	1.9012E−23	2.4128E−14	6.4541E−04	2.4913E−05	2.4231E+01	3.5822E−02	1.5097E−03
	Std	2.3203E−23	1.2388E−14	1.0455E−03	2.0841E−05	6.9385E−01	8.5951E−02	6.8587E−04
	Best	6.4334E−25	4.2800E−15	2.4941E−06	4.4154E−06	2.3289E+01	3.8894E−05	5.6648E−04
	Worst	1.2638E−22	6.5414E−14	6.4816E−03	1.2927E−04	2.8218E+01	2.5785E−01	4.7631E−03
HPSOBOA	Mean	3.3669E−176	3.3248E−88	1.7604E−176	7.0753E−88	5.7188E+00	1.3166E−04	8.2205E−03
	Std	1.5419E−175	3.2136E−88	8.1826E−176	2.0586E−87	2.9243E+01	5.7215E+00	7.6220E−05
	Best	0.0000E+00	1.7935E−88	0.0000E+00	4.8975E−89	3.5345E−01	5.4829E−01	6.9237E−05
	Worst	3.5755E−178	6.0323E−89	2.0794E−178	1.9018E−87	2.8943E+01	4.0940E+00	1.4659E−06
ISSWOA	Mean	1.1400E−193	1.2915E−116	1.5700E−183	5.3000E−105	8.4514E−04	2.7557E−06	7.2567E−05
	Std	0.0000E+00	1.2900E−115	0.0000E+00	5.3000E−104	8.2669E−04	9.2957E−07	6.5008E−05
	Best	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	8.7615E−08	1.0100E−06	7.0500E−07
	Worst	1.1400E−191	1.2900E−114	1.5700E−181	5.3000E−103	4.1000E−03	5.7600E−06	4.6813E−04

5.5 Analysis of ISSWOA for standard multimodal benchmark functions

Table 6 shows the test results of ISSWOA and other algorithms on multimodal benchmark functions. Compared with SSA, ISSA demonstrated a greater ability to escape local optima under the influence of Levy flight mechanism, as reflected in the mean of F8 and the mean and optimal value of F12. For IWOA, the improved spiral update mechanism conferred a good global search capability; therefore, the optimal values of F12 and F13 found by IWOA were approximately 10^{-3} times the optimal values found by WOA. ISSWOA inherited the advantages of IWOA and ISSA, had good performance, and was superior to the other algorithms, especially on F12 and F13.

Figures 14 and 15 display the iteration curves and box diagrams of ISSWOA and the other algorithms on standard multimodal benchmark functions. It can be seen from Fig. 14 that ISSWOA still performed better than the other algorithms in terms of convergence speed for the standard multimodal benchmark functions. From Fig. 15, except for F8, the box plot of ISSWOA resembled a straight line, indicating that ISSWOA had good robustness on the standard multimode benchmark function.

5.6 Analysis of ISSWOA for standard fixed-dimensional multimodal benchmark functions

Table 7 shows the test results of ISSWOA and other algorithms on standard fixed-dimensional multimodal benchmark functions. Compared with SSA, except for F16 and F17, ISSA escaped the local optimum and found the global optimum under the influence of Levy flight mechanism. For F17, the mean value of ISSA was better, indicating that ISSA had stronger robustness. Under the influence of the improved spiral update mechanism, IWOA found more global optimal solutions than WOA, but because IWOA could not escape from the local optimum, its optimization stability was insufficient. Under the influence of Levy flight mechanism, ISSWOA's optimization performance was more stable compared with other algorithms. ISSWOA performed the best except for F1; however, ISSWOA still found its global optimal solution, and its performance is better than some other algorithms.

It can be seen from Fig. 16 that the iteration speed of ISSWOA on standard fixed-dimensional multimodal benchmark functions was optimal except for F17, F18, and F21. It can be seen from Fig. 17 that, although the ISSWOA boxplot was not as good as performance on standard unimodal benchmark functions and standard multimodal benchmark functions, the distribution of ISSWOA 100 times results was the most concentrated and thus the overall performance was better.

5.7 Analysis on the optimization efficiency of ISSWOA

The efficiency of finding the optimal solution is also an important indicator of algorithm performance. This information is represented in Tables 8 and 9, which record average time required for 300 iterations and the number of iterations

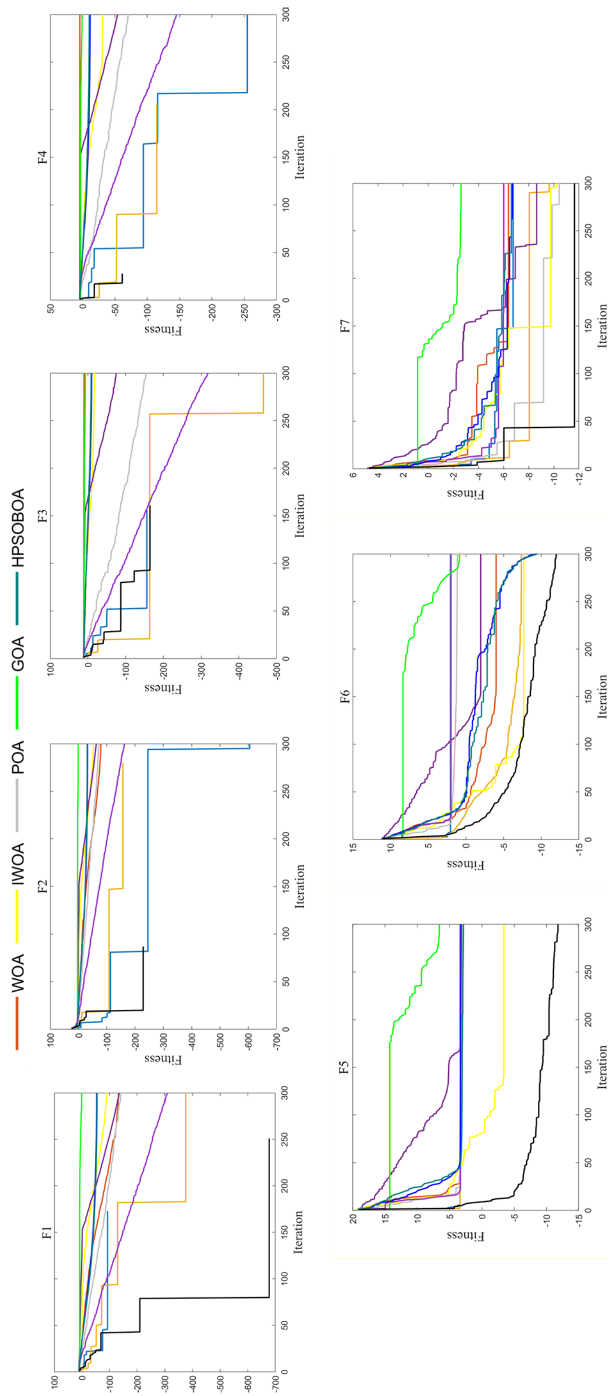

Fig. 12 Iteration curve of ISSWOA and other algorithms for standard unimodal benchmark functions

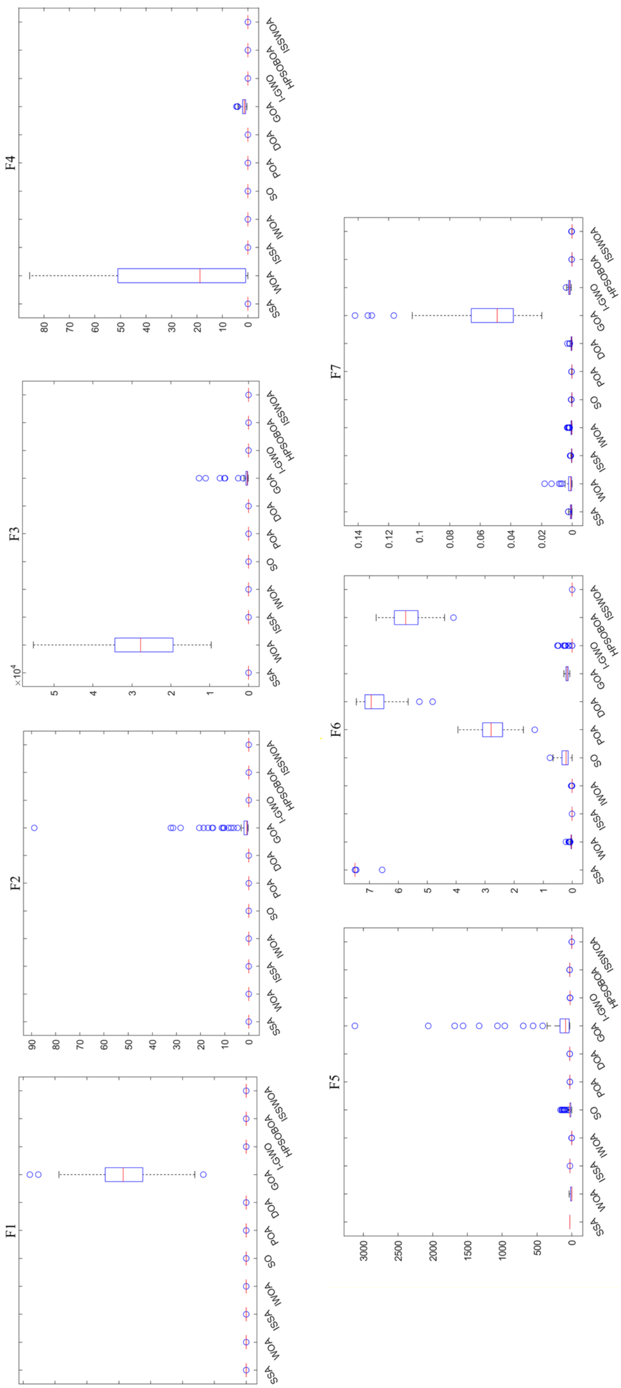

Fig. 13 Box plot of ISSWOA and other algorithms for standard unimodal benchmark functions

Table 6 Comparative result of ISSWOA with other algorithms for standard multimodal benchmark functions

Algorithms	Parameters	F8	F9	F10	F11	F12	F13
SSA	Mean	−5.0798E+03	0.0000E+00	8.8800E−16	0.0000E+00	1.6508E+00	7.2334E−04
	Std	1.7561E+03	0.0000E+00	9.9104E−32	0.0000E+00	7.6635E−02	1.2298E−03
	Best	−9.4550E+03	0.0000E+00	8.8800E−16	0.0000E+00	1.1254E+00	2.4619E−04
	Worst	−2.5966E+03	0.0000E+00	8.8800E−16	0.0000E+00	1.6690E+00	1.2831E−02
WOA	Mean	−1.1260E+04	5.6800E−16	4.4041E−15	3.5854E−03	3.7951E−03	1.2162E−01
	Std	1.4448E+03	5.6800E−15	2.2285E−15	3.5854E−02	3.5557E−03	9.7802E−02
	Best	−1.2569E+04	0.0000E+00	8.8800E−16	0.0000E+00	6.4192E−04	7.2709E−03
	Worst	−8.2880E+03	5.6800E−14	7.9900E−15	3.5854E−01	2.1031E−02	5.1593E−01
ISSA	Mean	−8.4522E+03	0.0000E+00	8.8818E−16	0.0000E+00	1.0507E−04	5.2747E−04
	Std	6.6582E+02	0.0000E+00	3.9642E−31	0.0000E+00	3.0463E−05	2.4129E−03
	Best	−1.0744E+04	0.0000E+00	8.8818E−16	0.0000E+00	4.6575E−05	1.0882E−04
	Worst	−6.5775E+03	0.0000E+00	8.8818E−16	0.0000E+00	1.8455E−04	1.1352E−02
IWOA	Mean	−1.2381E+04	0.0000E+00	8.8800E−16	0.0000E+00	8.0899E−05	1.2270E−03
	Std	7.8932E+02	0.0000E+00	9.9104E−32	0.0000E+00	3.7182E−04	2.7117E−03
	Best	−1.2569E+04	0.0000E+00	8.8800E−16	0.0000E+00	2.1000E−07	2.8800E−06
	Worst	−7.3766E+03	0.0000E+00	8.8800E−16	0.0000E+00	3.7000E−03	1.8000E−02
SO	Mean	−1.2413E+04	7.1375E+00	4.4054E−15	1.3148E−02	1.1983E−01	5.2431E−01
	Std	2.1662E+02	9.7841E+00	3.5527E−16	3.3213E−02	1.9076E−01	5.6829E−01
	Best	−1.2569E+04	0.0000E+00	8.8818E−16	0.0000E+00	1.9573E−05	8.9506E−05
	Worst	−1.1640E+04	3.5897E+01	4.4409E−15	1.7054E−01	9.7856E−01	2.6147E+00
POA	Mean	−7.8812E+03	0.0000E+00	3.1974E−15	0.0000E+00	1.5140E−01	2.5329E+00
	Std	5.9747E+02	0.0000E+00	1.7031E−15	0.0000E+00	6.6778E−02	5.1124E−01
	Best	−9.4127E+03	0.0000E+00	8.8818E−16	0.0000E+00	5.1076E−02	1.4635E+00
	Worst	−6.3266E+03	0.0000E+00	4.4409E−15	0.0000E+00	3.2385E−01	2.9839E+00
DOA	Mean	−5.2315E+03	8.4455E−02	8.8818E−16	0.0000E+00	1.0621E+00	2.9812E+00
	Std	1.1788E+03	1.8786E−01	0.0000E+00	0.0000E+00	2.8098E−01	1.2829E−01
	Best	−9.0047E+03	0.0000E+00	8.8818E−16	0.0000E+00	3.6548E−01	1.7435E+00
	Worst	−2.7189E+03	9.4258E−01	8.8818E−16	0.0000E+00	1.6043E+00	2.9999E+00
GOA	Mean	−7.1085E+03	1.1778E+02	2.0765E+00	1.6461E−01	2.2921E+00	7.9238E−02
	Std	7.3827E+02	4.8550E+01	7.7220E−01	5.6963E−02	1.3165E+00	4.7475E−02
	Best	−9.1736E+03	3.3830E+01	2.1240E−01	6.9528E−02	3.4125E−01	1.6924E−02
	Worst	−5.1286E+03	2.5293E+02	4.3614E+00	3.6282E−01	7.1621E+00	2.7098E−01
I-GWO	Mean	−8.3878E+03	2.3838E+01	1.0009E−12	3.5453E−03	5.0088E−03	5.4391E−02
	Std	1.8796E+03	2.6883E+01	5.7137E−13	6.7634E−03	1.8081E−02	6.9264E−02
	Best	−1.1249E+04	5.0845E+00	1.9273E−13	0.0000E+00	2.9595E−06	4.1754E−05
	Worst	−5.0495E+03	1.7258E+02	3.3760E−12	2.9529E−02	1.1008E−01	2.0896E−01
HPSOBOA	Mean	−3.6752E+03	4.6947E−01	8.8818E−16	0.0000E+00	7.4102E−01	2.9680E+00
	Std	4.2768E+02	1.6840E+00	3.9642E−31	0.0000E+00	1.3795E−01	1.2568E−01
	Best	−4.7303E+03	0.0000E+00	8.8818E−16	0.0000E+00	3.4795E−01	2.4073E+00
	Worst	−2.8894E+03	1.4461E+01	8.8818E−16	0.0000E+00	1.0245E+00	3.0943E+00
ISSWOA	Mean	−1.2415E+04	0.0000E+00	8.8818E−16	0.0000E+00	1.9933E−07	3.5996E−07
	Std	5.3394E+02	0.0000E+00	0.0000E+00	0.0000E+00	7.5053E−08	3.1410E−07
	Best	−1.2569E+04	0.0000E+00	8.8818E−16	0.0000E+00	4.4900E−08	8.9100E−08
	Worst	−8.8384E+03	0.0000E+00	8.8818E−16	0.0000E+00	4.2400E−07	3.0900E−06

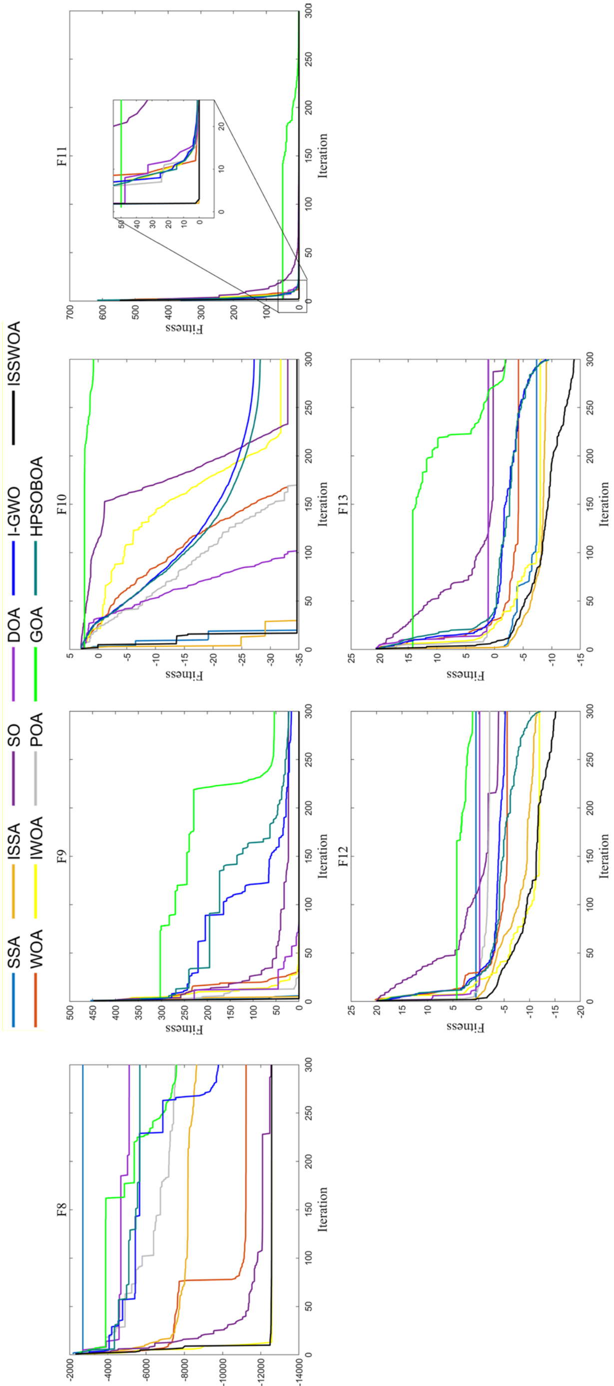

Fig. 14 Iteration curve of ISSWOA and other algorithms for standard multimodal benchmark functions

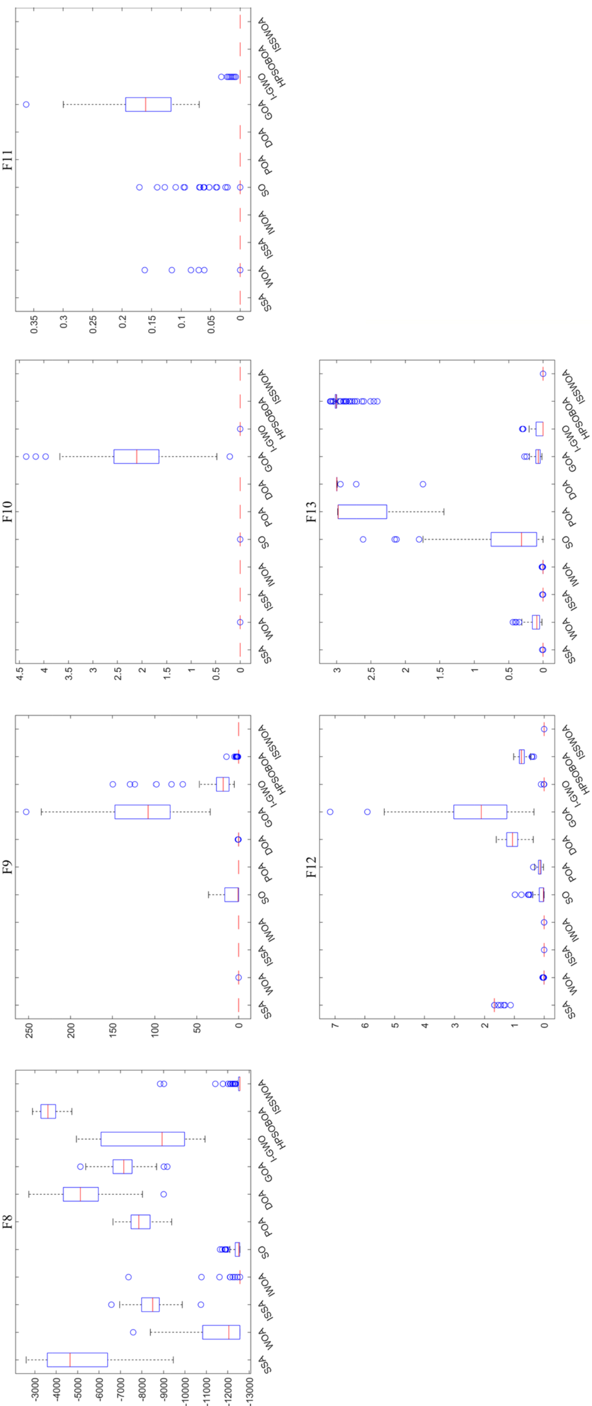

Fig. 15 Box plot of ISSWOA and other algorithms for standard multimodal benchmark functions

Table 7 Comparative result of ISSWOA with other algorithms for standard fixed-dimensional multimodal benchmark functions

Algorithms	Parameters	F14	F15	F16	F17	F18	F19	F20	F21	F22	F23
SSA	Mean	9.1516E+00	3.0962E−04	−1.0238E+00	3.9801E−01	1.1150E+01	−3.8546E+00	−3.2829E+00	−5.1328E+00	−5.2588E+00	−5.3412E+00
	Std	4.0344E+00	1.6702E−06	9.0739E−03	6.3211E−04	1.0998E+01	6.4381E−03	4.7993E−02	4.3060E−01	8.9976E−01	1.0502E+00
	Best	9.9802E−01	3.0761E−04	−1.0316E+00	3.9789E−01	3.0095E+00	−3.8627E+00	−3.3214E+00	−8.2365E+00	−1.0331E+01	−1.0525E+01
	Worst	1.2671E+01	3.1599E−04	−9.8860E−01	4.0409E−01	7.0607E+01	−3.8353E+00	−3.0359E+00	−5.0533E+00	−5.0857E+00	−5.1262E+00
WOA	Mean	2.1509E+00	6.3763E−04	−1.0316E+00	3.9789E−01	4.3545E+00	−3.8452E+00	−3.1844E+00	−1.0076E+01	−1.0337E+01	−1.0241E+01
	Std	2.2060E+00	4.9892E−04	1.0000E−10	2.8276E−05	5.9340E+00	2.9375E−02	8.0256E−02	1.2122E−01	9.5728E−02	1.1828E+00
	Best	9.9800E−01	3.1043E−04	−1.0316E+00	3.9789E−01	3.0000E+00	−3.8628E+00	−3.3184E+00	−1.0153E+01	−1.0403E+01	−1.0536E+01
	Worst	1.0763E+01	2.2519E−03	−1.0316E+00	3.9816E−01	3.0320E+01	−3.7431E+00	−2.9642E+00	−9.4496E+00	−9.6157E+00	−2.3832E+00
ISSA	Mean	2.6445E+00	3.1006E−04	−1.0316E+00	3.9789E−01	3.0000E+00	−3.8628E+00	−3.2839E+00	−5.8198E+00	−5.9380E+00	−6.2099E+00
	Std	3.9247E+00	2.9083E−06	6.6949E−16	1.1158E−16	0.0000E+00	2.6780E−15	5.9746E−02	1.8292E+00	1.9581E+00	2.1737E+00
	Best	9.9800E−01	3.0750E−04	−1.0316E+00	3.9789E−01	3.0000E+00	−3.8628E+00	−3.3220E+00	−1.0153E+01	−1.0403E+01	−1.0536E+01
	Worst	1.2671E+01	3.2284E−04	−1.0316E+00	3.9789E−01	3.0000E+00	−3.8628E+00	−3.2031E+00	−5.0552E+00	−5.0876E+00	−5.1284E+00
IWOA	Mean	1.3148E+00	5.5684E−04	−1.0316E+00	3.9789E−01	3.5400E+00	−3.8581E+00	−3.2159E+00	−1.0148E+01	−1.0399E+01	−1.0533E+01
	Std	9.2977E−01	5.5705E−04	4.4843E−09	3.0374E−07	3.7990E+00	1.8332E−02	9.3598E−02	8.2426E−03	5.6367E−03	4.2451E−03
	Best	9.9800E−01	3.0800E−04	−1.0316E+00	3.9789E−01	3.0000E+00	−3.8628E+00	−3.3219E+00	−1.0153E+01	−1.0403E+01	−1.0536E+01
	Worst	5.9288E+00	2.2519E−03	−1.0316E+00	3.9789E−01	3.0000E+01	−3.7576E+00	−2.9096E+00	−1.0086E+01	−1.0370E+01	−1.0515E+01
SO	Mean	9.9811E−01	5.9162E−04	−1.0316E+00	3.9789E−01	3.0000E+00	−3.8628E+00	−3.3149E+00	−9.9007E+00	−1.0049E+01	−1.0369E+01
	Std	1.0318E−03	2.0000E−03	0.0000E+00	1.1158E−16	0.0000E+00	1.7853E−15	2.8378E−02	7.4295E−01	1.0020E+00	5.3022E−01
	Best	9.9800E−01	3.0749E−04	−1.0316E+00	3.9789E−01	3.0000E+00	−3.8628E+00	−3.3220E+00	−1.0153E+01	−1.0403E+01	−1.0536E+01
	Worst	1.0083E+00	2.0363E−02	−1.0316E+00	3.9789E−01	3.0000E+00	−3.8628E+00	−3.2031E+00	−5.3526E+00	−5.0741E+00	−6.8592E+00
POA	Mean	9.9800E−01	3.0751E−04	−1.0316E+00	3.9789E−01	3.0000E+00	−3.8628E+00	−3.3000E+00	−1.0101E+01	−1.0403E+01	−1.0536E+01
	Std	4.9901E−17	1.5158E−07	6.3120E−16	0.0000E+00	1.0951E−15	2.1204E−15	5.9417E−02	5.0976E−01	1.0857E−03	3.9081E−05
	Best	9.9800E−01	3.0749E−04	−1.0316E+00	3.9789E−01	3.0000E+00	−3.8628E+00	−3.3220E+00	−1.0153E+01	−1.0403E+01	−1.0536E+01
	Worst	9.9800E−01	3.0879E−04	−1.0316E+00	3.9789E−01	3.0000E+00	−3.8628E+00	−3.1320E+00	−5.0552E+00	−1.0392E+01	−1.0536E+01
DOA	Mean	3.2984E+00	7.9804E−03	−1.0302E+00	3.9884E−01	3.0003E+00	−3.8694E+00	−3.1209E+00	−9.4820E+00	−9.5257E+00	−9.7386E+00
	Std	2.7311E+00	1.0962E−02	2.8618E−03	1.6125E−03	3.0067E−04	5.1720E−02	1.5385E−01	5.7247E−01	7.9196E−01	8.1833E−01
	Best	9.9800E−01	3.3931E−04	−1.0316E+00	3.9789E−01	3.0000E+00	−3.8623E+00	−3.3126E+00	−1.0112E+01	−1.0390E+01	−1.0438E+01

Table 7 (continued)

Algorithms	Parameters	F14	F15	F16	F17	F18	F19	F20	F21	F22	F23
	Worst	1.0764E+01	7.6862E−02	− 1.0157E+00	4.0556E−01	3.0014E+00	− 3.5890E+00	− 2.3920E+00	− 7.4170E+00	− 4.6378E+00	− 4.8209E+00
GOA	Mean	9.9800E−01	6.5617E−03	− 1.0316E+00	3.9801E−01	3.0000E+00	− 3.8628E+00	− 3.2649E+00	− 7.5628E+00	− 7.2914E+00	− 7.7440E+00
	Std	3.1789E−15	9.7364E−03	9.4706E−12	9.1121E−04	8.5793E−11	4.8428E−06	5.9749E−02	3.1600E+00	3.5546E+00	3.4439E+00
	Best	9.9800E−01	3.1009E−04	− 1.0316E+00	3.9789E−01	3.0000E+00	− 3.8628E+00	− 3.3220E+00	− 1.0153E+01	− 1.0403E+01	− 1.0536E+01
	Worst	9.9800E−01	5.6543E−02	− 1.0316E+00	4.0663E−01	3.0000E+00	− 3.8627E+00	− 3.2020E+00	− 2.6305E+00	− 1.8376E+00	− 2.4217E+00
I-GWO	Mean	9.9800E−01	3.1393E−04	− 1.0316E+00	3.9789E−01	3.0000E+00	− 3.8628E+00	− 3.3150E+00	− 1.0132E+01	− 1.0403E+01	− 1.0536E+01
	Std	2.7241E−16	4.1212E−05	2.0487E−15	5.4244E−11	2.2613E−15	1.2800E−14	2.6881E−02	2.0908E−01	2.1315E−04	1.5970E−04
	Best	9.9800E−01	3.0749E−04	− 1.0316E+00	3.9789E−01	3.0000E+00	− 3.8628E+00	− 3.3220E+00	− 1.0153E+01	− 1.0403E+01	− 1.0536E+01
	Worst	9.9800E−01	6.6055E−04	− 1.0316E+00	3.9789E−01	3.0000E+00	− 3.8628E+00	− 3.2031E+00	− 8.0621E+00	− 1.0401E+01	− 1.0535E+01
HPSOBOA	Mean	2.5841E+00	9.7030E−04	− 1.0311E+00	4.3680E−01	4.3758E+00	− 3.8138E+00	− 3.1439E+00	− 2.4661E+00	− 2.5753E+00	− 2.5315E+00
	Std	1.3933E+00	6.0168E−04	7.4246E−04	3.5772E−02	1.6424E+00	7.1858E−02	1.1599E−01	4.8947E−01	9.9740E−01	6.8775E−01
	Best	9.9800E−01	3.6720E−04	− 1.0316E+00	3.9825E−01	3.0128E+00	− 3.8627E+00	− 3.3081E+00	− 4.0688E+00	− 9.3268E+00	− 6.1314E+00
	Worst	7.0094E+00	4.9350E−03	− 1.0285E+00	5.7092E−01	1.4253E+01	− 3.5557E+00	− 2.7544E+00	− 1.9924E+00	− 1.8275E+00	− 1.6805E+00
ISSWOA	Mean	1.7113E+00	3.0749E−04	− 1.0316E+00	3.9789E−01	3.0000E+00	− 3.8628E+00	− 3.3208E+00	− 1.0153E+01	− 1.0403E+01	− 1.0536E+01
	Std	2.5024E+00	1.3633E−09	0.0000E+00	2.9319E−10	0.0000E+00	7.0345E−10	1.2012E−02	3.7524E−08	7.1412E−15	1.7853E−15
	Best	9.9800E−01	3.0749E−04	− 1.0316E+00	3.9789E−01	3.0000E+00	− 3.8628E+00	− 3.3220E+00	− 1.0153E+01	− 1.0403E+01	− 1.0536E+01
	Worst	1.2671E+01	3.0749E−04	− 1.0316E+00	3.9789E−01	3.0000E+00	− 3.8628E+00	− 3.2019E+00	− 1.0153E+01	− 1.0403E+01	− 1.0536E+01

required to find the optimal solution for each benchmark function, respectively. It can be seen from Table 8 that, although ISSWOA was composed of ISSA and IWOA, its running time was short, and on some benchmark functions, its running time was even less than that of its constituent algorithms. Its running time is similar to that of I-GWO and HPSOBOA, which are both improved algorithms. From the number of iterations required for each algorithm to find its optimal solution over 100 trials, shown in Table 9, ISSWOA required fewer iterations to find the global optimal solution on most benchmark functions, compared with some other algorithms. Considering the number of iterations and time required, ISSWOA is a competitive proposal in optimization efficiency.

5.8 Wilcoxon rank sum test

A Wilcoxon rank sum statistical test with 5% accuracy was used to investigate the differences between ISSWOA and other algorithms. The results are shown in Table 10, There were significant differences between ISSWOA and the other algorithms in the test results of most benchmark functions. Combined with Tables 5, 6, and 7, it can be seen that ISSWOA had significant advantages over other algorithms in most cases. Therefore, ISSWOA displayed strong performance in the optimization of three types of standard benchmark functions.

6 Application of ISSWOA to engineering problems

Engineering problems are complex, nonlinear optimization problems. Consequently, the performance of optimization algorithms is best evaluated by applying them to practical problems (rather than theoretically). For the simulations, we set the number of iterations as 500 and the population size as 100. The optimization results for 30 independent runs of ISSWOA and similar algorithms were compared, with the number of discoverers set as 40 and the number of guards as 30. The comparison algorithm is shown in Table 11.

6.1 Treatment of constraints

For engineering problems with constraints, the punishment function [61] is used to amplify the fitness of solutions according to the number of exceeded constraints. When the solution does not meet the constraint conditions, the equation for fitness is

$$F = f + \sum_{1}^{i} \left(\text{lam} * g_i^2 * Q_i \right), \tag{16}$$

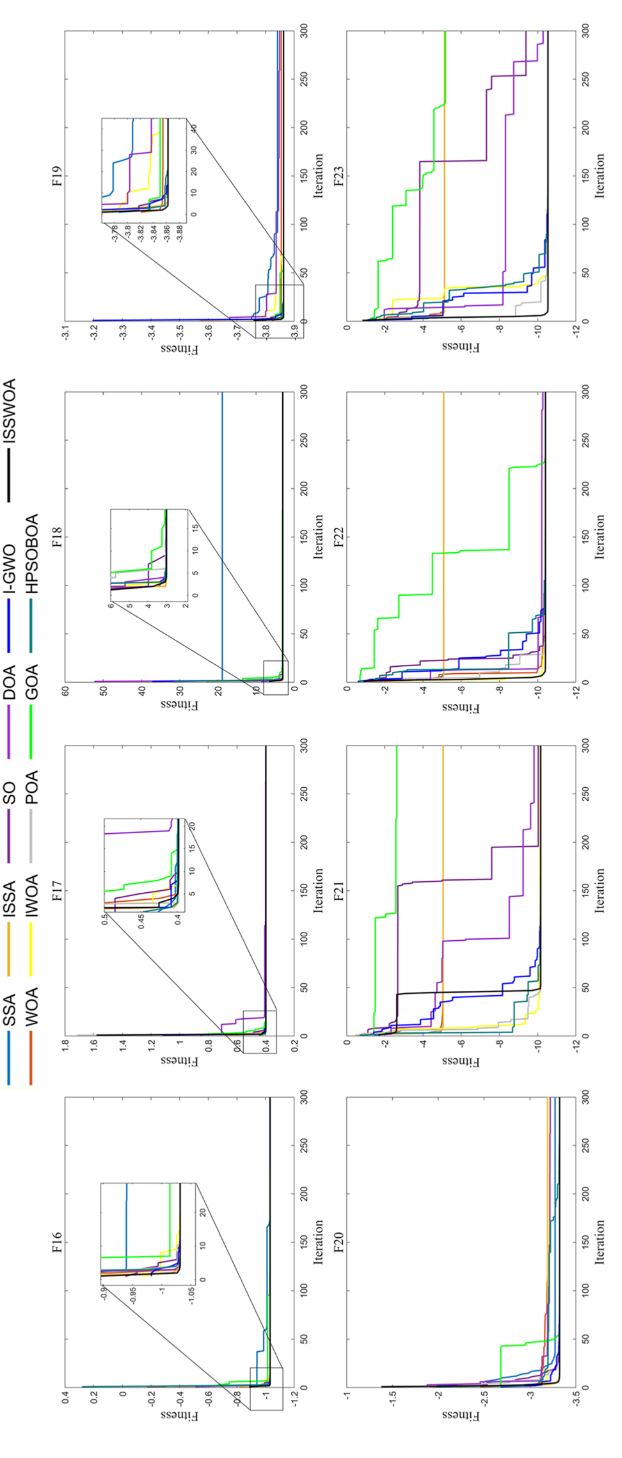

Fig. 16 Iteration curve of ISSWOA and other algorithms for fixed-dimensional multimodal benchmark functions

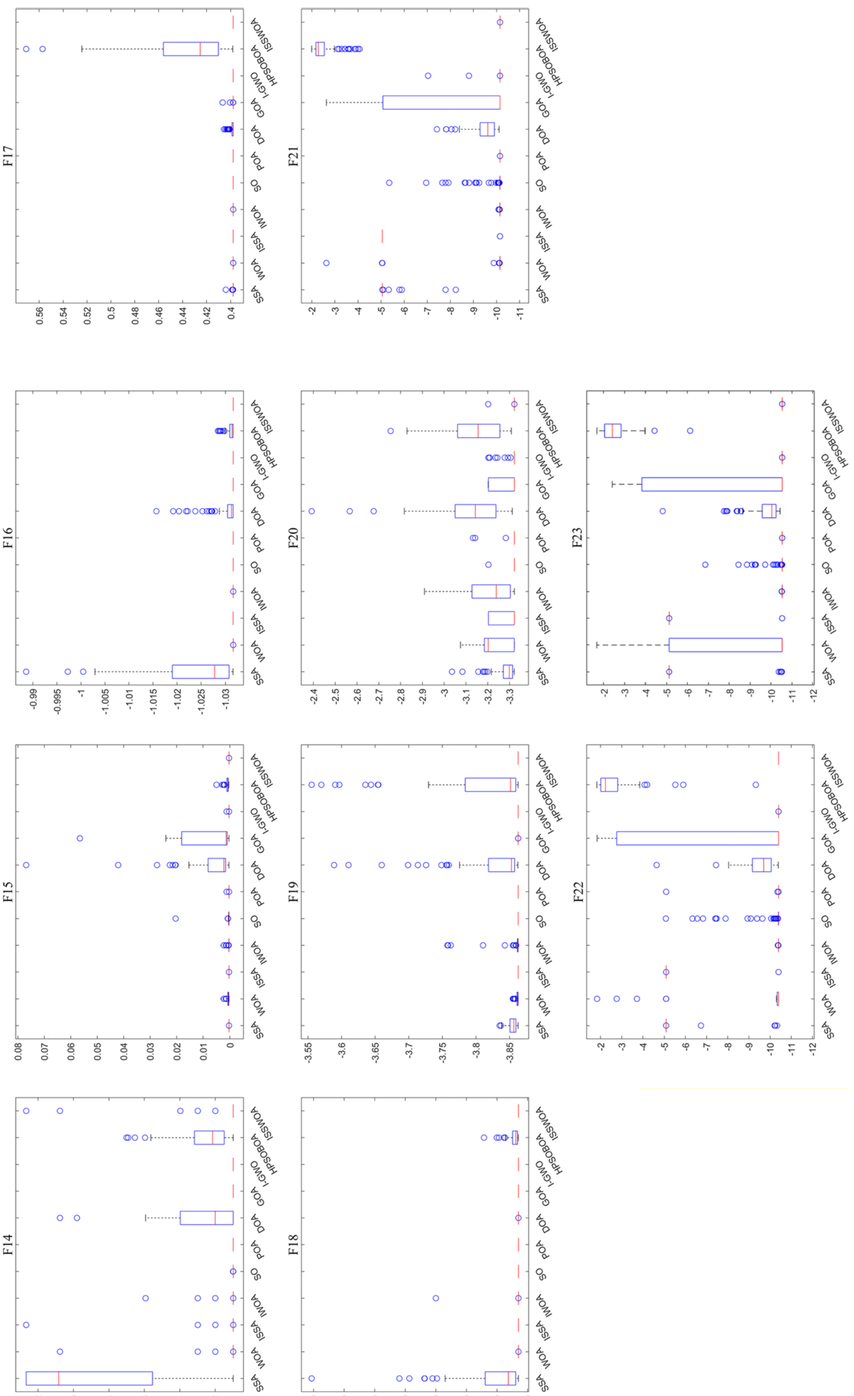

Fig. 17 Box plot of ISSWOA and other algorithms for fixed-dimensional multimodal benchmark functions

Table 8 Wall-clock time costs of ISSWOA and other algorithms on 23 benchmarks

Function	SSA	WOA	ISSA	IWOA	SO	POA	DOA	GOA	I-GWO	HPSOBOA	ISSWOA
F1	1.4254	0.2024	0.9923	0.1981	0.2340	0.3058	0.4646	150.1456	1.0257	0.3377	0.9196
F2	1.2315	0.1188	0.8137	0.1253	0.1388	0.1484	0.2259	149.3986	0.8481	1.0190	0.5912
F3	1.2408	0.0921	0.7854	0.1036	0.1184	0.1103	0.1824	148.7655	0.8386	0.2130	0.5111
F4	1.2607	0.0999	0.7902	0.1082	0.1131	0.1260	0.1719	150.6561	0.8316	0.1690	0.5650
F5	1.2695	0.1194	0.8178	0.1308	0.1339	0.1666	0.2203	150.2642	0.8853	0.1883	0.5922
F6	1.8456	0.4073	1.4071	0.4084	0.4085	0.6840	1.0770	153.7409	1.4328	0.4537	1.7304
F7	1.8803	0.4011	1.3849	0.3888	0.3935	0.6728	0.9764	150.8541	1.4202	0.4388	1.7269
F8	1.3314	0.6527	1.5017	0.6405	0.6071	1.1245	1.6929	10.9513	1.7558	0.9924	2.6754
F9	0.2123	0.0862	0.3278	0.0941	0.0533	0.0962	0.1273	20.5590	0.6457	0.1238	0.3843
F10	0.1903	0.0780	0.3035	0.0878	0.0478	0.0955	0.1234	10.3653	0.6255	0.4494	0.3764
F11	0.1781	0.0830	0.3083	0.1134	0.1022	0.0769	0.1261	10.8523	0.6598	0.1929	0.4081
F12	0.1441	0.0746	0.2869	0.0861	0.1036	0.0751	0.1093	10.4478	0.5986	0.1174	0.3499
F13	0.2225	0.1118	0.4160	0.1070	0.0902	0.1160	0.1722	20.3734	0.6779	0.7313	0.4187
F14	0.2813	0.0968	0.4037	0.1096	0.0929	0.1231	0.1650	30.4726	0.6906	0.6620	0.4424
F15	0.3034	0.1061	0.3876	0.1277	0.0850	0.1417	0.1968	20.2982	0.7041	1.1535	0.4998
F16	0.2888	0.1216	0.4169	0.1278	0.1071	0.1674	0.2235	21.3292	0.7497	1.0372	0.5038
F17	0.3178	0.1256	0.4433	0.1412	0.1115	0.2041	0.2684	20.2654	0.7505	1.1959	0.5731
F18	1.4254	0.2024	0.9923	0.1981	0.2340	0.3058	0.4646	150.1456	1.0257	0.3377	0.9196
F19	1.2315	0.1188	0.8137	0.1253	0.1388	0.1484	0.2259	149.3986	0.8481	1.0190	0.5912
F20	1.2408	0.0921	0.7854	0.1036	0.1184	0.1103	0.1824	148.7655	0.8386	0.2130	0.5111
F21	1.2607	0.0999	0.7902	0.1082	0.1131	0.1260	0.1719	150.6561	0.8316	0.1690	0.5650
F22	1.2695	0.1194	0.8178	0.1308	0.1339	0.1666	0.2203	150.2642	0.8853	0.1883	0.5922
F23	1.8456	0.4073	1.4071	0.4084	0.4085	0.6840	1.0770	153.7409	1.4328	0.4537	1.7304

Table 9 The number of iterations required for ISSWOA and other algorithms to find the optimal solution function

	SSA	WOA	ISSA	IWOA	SO	POA	DOA	GOA	I-GWO	HPSOBOA	ISSWOA
F1	1969	28,792	10,335	28,490	1126	8869	27,440	19,201	25,200	7026	12,160
F2	19,403	1498	15,900	21,263	20,992	2996	24,868	16,200	18,900	4501	5700
F3	7	135	26	176	2253	99	279	15,001	17,400	18	29
F4	23	1975	87	237	12,235	147	106	28,201	8400	126	20
F5	9	141	25	196	491	90	57	22,800	444	93	7
F6	11,710	6300	26,978	28,790	15,255	21,299	18,633	20,700	300	29,118	23,700
F7	29,097	8694	28,490	18,850	24,458	15,000	14,945	1800	300	16,524	14,700
F8	15,309	1191	299	300	116	286	1267	300	278	26,407	294
F9	263	6892	12,571	5238	27,452	286	20,400	19,200	6600	10,522	5400
F10	13,225	300	256	296	124	150	61	300	208	1	299
F11	11,869	296	139	15,300	4318	162	398	900	285	18,301	599
F12	13,202	9810	178	29,984	4322	249	249	300	240	7816	296
F13	15,992	19,484	254	14,975	27,738	201	19,163	600	277	15,301	299
F14	11,537	290	286	20,999	22,944	16,799	236	3000	22,151	13,501	10,196
F15	12,751	14,351	842	18,565	17,541	300	258	900	5395	29	22,500
F16	27,552	23,943	5341	283	27,441	1786	14,363	5400	5385	8234	10,762
F17	813	15,886	705	1597	438	564	20,507	3000	27,896	26,121	5998
F18	1969	28,792	10,335	28,490	1126	8869	27,440	19,201	25,200	7026	12,160
F19	19,403	1498	15,900	21,263	20,992	2996	24,868	16,200	18,900	4501	5700
F20	7	135	26	176	2253	99	279	15,001	17,400	18	29
F21	23	1975	87	237	12,235	147	106	28,201	8400	126	20
F22	9	141	25	196	491	90	57	22,800	444	93	7
F23	11,710	6300	26,978	28,790	15,255	21,299	18,633	20,700	300	29,118	23,700

Table 10 p-values of the Wilcoxon rank sum test with 5% significance for 23 functions

Function	SSA	WOA	ISSA	IWOA	SO	POA	DOA	GOA	I-GWO	HPSOBOA
F1	9.68E−03	6.30E−37	1.43E−04	6.30E−37	6.30E−37	6.30E−37	6.30E−37	6.30E−37	6.30E−37	6.30E−37
F2	6.25E−02	4.32E−35	9.19E−04	4.32E−35	4.32E−35	4.32E−35	4.32E−35	4.32E−35	4.32E−35	4.32E−35
F3	1.50E−04	1.23E−36	1.89E−02	1.23E−36	1.23E−36	1.23E−36	1.23E−36	1.23E−36	1.23E−36	1.23E−36
F4	1.42E−02	7.78E−35	1.03E−02	7.78E−35	7.78E−35	7.78E−35	7.78E−35	7.78E−35	7.78E−35	7.78E−35
F5	2.56E−34	2.56E−34	2.56E−34	2.56E−34	2.56E−34	2.56E−34	2.53E−34	2.56E−34	2.56E−34	2.56E−34
F6	1.04E−37	2.56E−34	2.56E−34	2.56E−34	2.56E−34	2.56E−34	2.56E−34	2.56E−34	2.56E−34	2.56E−34
F7	1.95E−29	1.60E−26	7.53E−22	6.66E−21	1.01E−15	5.33E−13	1.67E−20	2.56E−34	2.56E−34	9.80E−01
F8	1.02E−34	2.26E−20	1.05E−34	3.28E−18	4.79E−08	1.02E−34	1.02E−34	1.02E−34	1.42E−34	7.99E−35
F9	NaN	3.22E−01	NaN	NaN	8.15E−38	NaN	5.61E−18	5.64E−39	5.64E−39	6.07E−05
F10	NaN	2.36E−29	NaN	NaN	2.55E−44	9.69E−18	3.52E−45	5.64E−39	5.63E−39	3.52E−45
F11	NaN	3.22E−01	NaN	NaN	2.72E−06	NaN	NaN	5.64E−39	3.97E−07	NaN
F12	1.81E−37	2.56E−34	2.56E−34	1.32E−33	2.56E−34	2.56E−34	2.56E−34	2.56E−34	2.56E−34	2.56E−34
F13	2.56E−34	2.56E−34	2.56E−34	2.64E−34	2.56E−34	2.56E−34	2.54E−34	2.56E−34	2.56E−34	2.56E−34
F14	1.03E−31	1.79E−15	2.76E−15	1.79E−03	3.35E−02	2.04E−04	8.14E−16	2.04E−04	2.04E−04	1.18E−23
F15	7.77E−38	7.76E−38	7.76E−38	7.74E−38	8.85E−38	7.40E−01	7.69E−38	7.41E−38	4.80E−32	7.77E−38
F16	5.64E−39	3.22E−01	3.52E−45	2.57E−12	NaN	NaN	4.12E−39	NaN	NaN	3.62E−39
F17	4.27E−38	7.38E−22	5.47E−44	3.64E−37	1.33E−02	1.33E−02	3.89E−38	7.76E−15	1.33E−02	4.27E−38
F18	5.64E−39	1.12E−36	NaN	5.64E−39	NaN	NaN	6.60E−30	NaN	NaN	5.64E−39
F19	2.47E−35	2.47E−35	2.63E−40	2.47E−35	4.28E−13	4.28E−13	2.47E−35	2.44E−23	4.28E−13	2.47E−35
F20	4.07E−33	7.97E−34	4.81E−06	1.17E−33	7.73E−34	3.59E−33	6.28E−34	8.56E−13	5.00E−33	7.29E−34
F21	1.67E−34	1.67E−34	3.01E−37	1.67E−34	4.35E−06	4.27E−04	1.67E−34	8.48E−01	5.23E−34	1.67E−34
F22	5.64E−39	5.64E−39	4.06E−39	5.64E−39	9.08E−08	1.17E−01	5.64E−39	1.91E−01	2.99E−33	5.64E−39
F23	2.46E−34	2.46E−34	1.73E−36	2.46E−34	3.06E−10	1.94E−01	2.46E−34	8.38E−02	1.18E−32	2.46E−34

where F is the fitness value of introducing the penalty function, f is the fitness value of the unpunished function, lam is a constant greater than 1, g_i is the ith constraint value, and Q_i is a binary variable, equal to 1 when the current constraints do not meet the conditions (and zero otherwise).

6.2 Tension/compression spring design problem

As illustrated in Fig. 18, the tension/compression spring design problem [62] in mechanical engineering can be used to evaluate optimization algorithms for various parameters. The problem consists of minimizing the mass under certain constraints, including four constraints and three design variables. The design variables are average diameter d of the spring coil (x_2), diameter w of the spring wire (x_1), and number

Table 11 The algorithm for comparison

Algorithms	Abbreviation
Chaos sparrow search algorithm with logarithmic spiral and adaptive step [38]	CLSSA
Ant lion optimizer [39]	ALO
Spotted hyena optimizer [40]	SHO
Hybrid Harris Hawks pattern search algorithm [41]	HHO-PS
Hybrid Harris Hawks-Sine Cosine Algorithm [42]	hHHO-SCA
Constrained topographical global optimization [43]	C-ITGO
Cuckoo search and differential evolution [44]	CSDE
Nelder–Mead simplex search and particle swarm optimization [45]	NM–PSO
Backtracking biogeography-based optimization [46]	BBBO
Adaptive firefly algorithm [47]	AFA
Passing vehicle search [48]	PVS
Henry gas solubility optimization [49]	HGSO
Opposition-based sine cosine algorithm [50]	OBSCA
Hybridizing salp swarm algorithm with particle swarm optimization algorithm [51]	HSSAPSO
Mine blast algorithm [52]	MBA
Chameleon swarm algorithm [53]	CSA
Backtracking search algorithm combined ε-constrained with self-adaptive control way of ε value [54]	BSA-SAεε
Bat algorithm [55]	BA
Genetic algorithm [2]	GA
Particle Swarm optimization [7]	PSO
Sine cosine algorithm-Gray wolf optimization [56]	SCA-GWO
Moth-flame optimization [57]	MFO
Improved gray wolf optimization [58]	IGWO
Golden Jackal Optimization [59]	GJO
Hybrid particle swarm optimizer with mutation [60]	HPSOM
Hybrid particle swarm optimization with wavelet mutation [60]	HPSOWM
Hybrid gradient descent PSO [60]	HGPSO

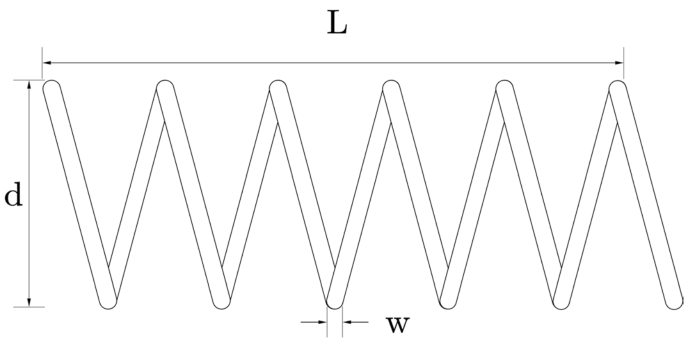

Fig. 18 Schematic of tension/compression spring design problem

Table 12 Best solutions obtained from ISSWOA and other algorithms for the tension/compression spring design problem

Algorithm	Design variable			Optimum cost
	w	*d*	*L*	
ISSWOA	0.0517	0.3577	11.2315	0.01266526
CLSSA	0.0518	0.3592	11.1441	0.0127
ALO	0.0517	0.3569	11.2793	0.0127
SHO	0.051144	0.343751	12.0955	0.012674
HHO-PS	0.051682	0.356552	11.29867	0.012665
C-ITGO	0.05168906	0.35671774	11.28896574	0.01266523
MBA	0.051689	0.356717	11.288965	0.012665

L of effective spring coils (x_3). Tables 12 and 13 compare the optimal solutions and constraints obtained by different algorithms. The statistical optimization results of the algorithms are given in Table 14. The problem is formulated as follows:

$$\min f\left(x_1, x_2, x_3\right) = \left(x_3 + 2\right)x_1^2 x_2 \tag{17}$$

subject to

$$g_1(X) = 1 - \frac{x_2^3 x_3}{71785 x_1^4} \le 0,$$

$$g_2(X) = \frac{x_2\left(4x_2 - x_1\right)}{12566 x_1^3\left(x_2 - x_1\right)} + \frac{1}{5108 x_1^2} - 1 \le 0,$$

$$g_3(X) = 1 - \frac{140.45 x_1}{x_2^2 x_3} \le 0,$$

$$g_4(X) = \frac{2\left(x_1 + x_2\right)}{3} - 1 \le 0.$$

Table 13 The constraints of ISSWOA and other algorithms for the tension/compression spring design problem

Algorithm	g_1	g_2	g_3	g_4
ISSWOA	− 4.4292E−07	− 1.2526E−08	− 4.0557	− 0.7270
CLSSA	0.0007	− 0.0004	− 4.0598	− 0.7260
ALO	0.0002	− 0.0002	− 4.0540	− 0.7276
SHO	− 0.0003	1.1591E−05	− 4.0258	− 0.7367
HHO-PS	− 1.1521E−05	9.2675E−06	− 4.0535	− 0.7278
C-ITGO	− 8.4449E−08	5.9200E−08	− 4.0538	− 0.7277
MBA	0.0000	0.0000	4.0522	0.7283

The variable ranges are $0.05 \leq x_1 \leq 2, 0.25 \leq x_2 \leq 1.3$, and $2.0 \leq x_3 \leq 15.0$.

6.3 Pressure vessel design problem

The pressure vessel design problem [63] illustrated in Fig. 19 is a typical hybrid optimization problem for minimizing the total cost while meeting production needs. This problem has four constraints and four design variables. The variables are: shell thickness *Th* (x_1), head thickness *Ts* (x_2), radius *R* (x_3), and container section length *L* (x_4). Tables 15 and 16 show the comparison among the optimal solutions and constraints obtained by different algorithms. The statistical optimization results of the algorithms are given in Table 17. The problem is formulated as follows:

$$\min f\left(x_1, x_2, x_3, x_4\right) = 0.6224x_1x_3x_4 + 1.781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3, \tag{18}$$

subject to

$$\begin{aligned} g_1(X) &= -x_1 + 0.0193x_3 \leq 0, \\ g_2(X) &= -x_2 + 0.00954x_3 \leq 0, \\ g_3(X) &= -\pi x_3^2x_4 - \frac{4}{3}\pi x_3^3 + 1296000 \leq 0, \\ g_4(X) &= x_4 - 240 \leq 0. \end{aligned}$$

Table 14 Comparison of statistical results using ISSWOA and other algorithm for the tension/compression spring design problem

Algorithm	Mean	Std	Best	Worst
ISSWOA	0.01274569	0.000371492	0.012665	0.012880
CLSSA	N/A	N/A	0.0127	N/A
ALO	N/A	N/A	0.0127	N/A
SHO	0.012684106	2.7E−05	0.012674	0.012715185
HHO-PS	0.012665	1.38E−09	0.012665	0.012665
C-ITGO	0.01266523	2.81E−9	0.01266523	0.01266525
MBA	0.012713	6.30E−05	0.012665	0.012900

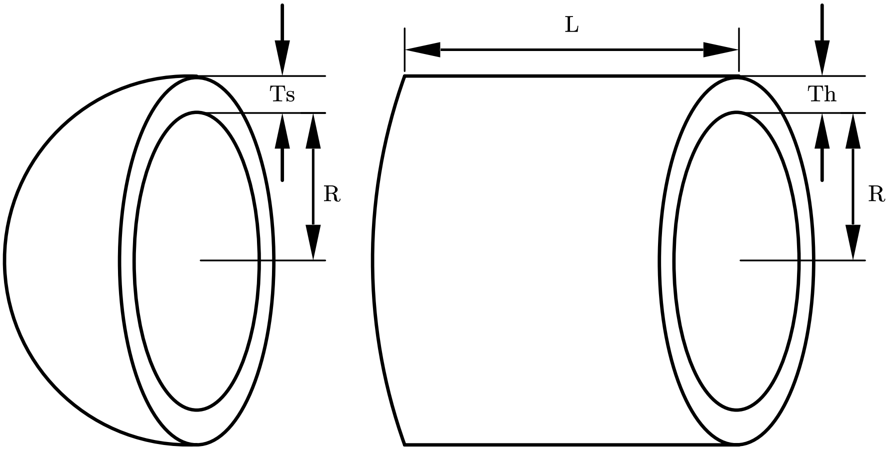

Fig. 19 Schematic of the pressure vessel design problem

Table 15 Best solutions obtained from ISSWOA and other algorithms for the pressure vessel design problem

Algorithm	Design variable				Optimum cost
	Th	*Ts*	*R*	*L*	
ISSWOA	0.778482	0.384922	40.335690	199.777495	5886.2529
hHHO-SCA	0.945909	0.447138	46.8513	125.4684	6393.0927
CSDE	0.812500	0.437500	42.1000	176.6000	6059.7100
C-ITGO	13.0	7.0	41.5984	176.6366	6059.7143
NM–PSO	0.8036	0.3972	41.6392	182.4120	5930.3137
BBBO	1.1250	0.6250	58.1967	44.2721	7206.6400
MBA	0.7802	0.3856	40.4292	198.4964	5889.3216

The variable ranges are $0 \le x_1 \le 99, 0 \le x_2 \le 99, 10 \le x_3 \le 200$, and $10 \le x_4 \le 200$.

6.4 Speed reducer design problem

Figure 20 illustrates the speed reducer design problem [64]. It is an important and complex multi-constraint optimization problem in mechanical systems and consists of 7 variables and 11 constraints. The variables are tooth surface width B (x_1), gear modulus M (x_2), number Z of teeth in pinions (x_3), length l_1 of the first shaft between bearings (x_4), length l_2 of the second shaft between bearings (x_5), diameter d_1 of the first shaft (x_6), and diameter d_2 of the second shaft (x_7). Tables 18 and 19 compare the optimal solutions and constraints obtained by different algorithms. The statistical optimization results of the algorithms are given in Table 20. The problem is formulated as follows:

Table 16 The constraints of ISSWOA and other algorithms for the pressure vessel design problem

Algorithms	g_1	g_2	g_3	g_4
ISSWOA	0.000000	− 1.2526E−08	− 4.055723	− 0.727045
hHHO-SCA	0.000688	− 0.000432	− 4.059806	− 0.726000
CSDE	0.000171	− 0.000182	− 4.054021	− 0.727600
C-ITGO	− 0.000330	1.1591E−05	− 4.025794	− 0.736737
NM–PSO	− 1.1521E−05	9.2675E−06	− 4.053450	− 0.727844
BBBO	− 8.4449E−08	5.9200E−08	− 4.053786	− 0.727729
MBA	0.000000	0.000000	4.052248	0.728268

Table 17 Comparison of statistical results using ISSWOA and other algorithm for the pressure vessel design problem

Algorithm	Average	Std	Best	Worst
ISSWOA	5900.46016429125	456.198715837763	5886.1085	5914.62392647415
hHHO-SCA	N/A	N/A	6393.0927	N/A
CSDE	6261.417810301860	263.675819344899	6059.713339601150	1.52848528822763E+22
C-ITGO	6059.7143	9.8E−13	6059.7143	6059.7143
NM–PSO	5946.7901	9.1614	5930.3137	5960.0557
BBBO	N/A	N/A	7206.6400	N/A
MBA	6200.64765	160.34	5889.3216	6392.5062

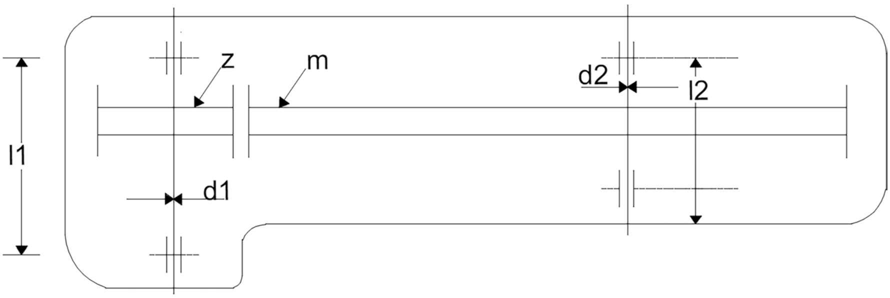

Fig. 20 Schematic of the speed reducer design problem

$$\begin{aligned} \min f(x_1,x_2,x_3,x_4,x_5,x_6,x_7) = {} & 0.7854x_1x_2^2\left(3.3333x_3^2 + 14.9334x_3 - 43.0934\right) \\ & - 1.508x_1\left(x_6^2 + x_7^2\right) + 7.4777\left(x_6^3 + x_7^3\right) \\ & + 0.7854\left(x_4x_6^2 + x_5x_7^2\right) \end{aligned} \tag{19}$$

subject to

Table 18 Comparison of statistical results using ISSWOA and other algorithm for the speed reducer design problem

Algorithm	Design variable							Optimum cost
	b	m	z	l_1	l_2	d_1	d_2	
ISSWOA	3.5000	0.7000	17.0000	7.3002	7.7160	3.3506	5.2867	2994.48
AFA	3.5	0.7	17	7.30248	7.80006	3.35021	5.286683	2996.372698
PVS	3.5	0.7	17	7.3	7.8	3.35021	5.28668	2996.34817
HGSO	3.498	0.71	17.02	7.67	7.810	3.36	5.289	2997.10
SHO	3.50159	0.7	17	7.3	7.8	3.35127	5.28874	2998.5507
OBSCA	3.0879	0.7550	26.4738	7.3650	7.9577	3.4950	5.2312	3056.3122
MBA	3.50000	0.7000	17.000	7.30003	7.71577	3.35022	5.286654	2994.48

Table 19 The constraints of ISSWOA and other algorithms for the speed reducer design problem

Constraints	AFA	PVS	HGSO	SHO	OBSCA	MBA	ISSWOA
g_1	−2.1550	−2.1550	−3.0121	−2.1682	−19.5987	−2.1550	−2.1553
g_2	−98.1350	−98.1349	−113.3056	−98.3602	−836.1436	−98.1350	−98.1403
g_3	−1.9197	−1.9236	−1.4834	−1.9285	−5.5351	−1.9236	−1.9251
g_4	−17.6578	−17.6582	−17.9199	−17.6888	−27.7734	−18.3064	−18.3052
g_5	0.3217	0.3260	−8.7947	−0.7177	−134.6255	−0.3182	−0.0612
g_6	0.0001	0.0016	−1.1343	−0.9913	26.9151	−0.0003	−0.0109
g_7	−28.1000	−28.1000	−27.9158	−28.1000	−20.0123	−28.1000	−28.1000
g_8	0.0000	0.0000	0.0732	−0.0023	0.9101	0.0000	−0.0001
g_9	−7.0000	−7.0000	−7.0732	−6.9977	−7.9101	−7.0000	−6.9999
g_{10}	−0.3772	−0.3747	−0.7300	−0.3731	−0.2225	−0.3747	−0.3743
g_{11}	−0.0847	−0.0847	−0.0921	−0.0824	−0.3034	−0.0005	−0.0006

Table 20 Comparison of statistical results using ISSWOA and other algorithm for the speed reducer design problem

Algorithm	Mean	Std	Best	Worst
ISSWOA	2997.8167279	2.269687	2994.485307	3001.902741
AFA	2996.514874	0.09	2996.372698	2996.669016
PVS	2996.348165	N/A	2996.348165	2996.348165
HGSO	2996.4	4.39E−05	2997.1	2996.9
SHO	2999.187	1.93193	2998.5507	3003.889
OBSCA	N/A	N/A	3056.3122	N/A
MBA	2996.769019	1.56	2994.482453	2999.652444

The variable ranges are $2.6 \leq x_1 \leq 3.6, 0.7 \leq x_2 \leq 0.8, 17 \leq x_3 \leq 28, 7.3 \leq x_4 \leq 8.3,$ $7.3 \leq x_5 \leq 8.3, 2.9 \leq x_6 \leq 3.9$, and $5 \leq x_7 \leq 5.5$.

6.5 Rolling element-bearing design problem

As illustrated in Fig. 21, the rolling element-bearing design problem consists of maximizing the dynamic bearing capacity of rolling bearings [65]. There are 10 decision variables in this problem: pitch diameter (D_m), ball diameter (D_b), number of balls (Z), inner raceway curvature coefficients (f_i), and outer raceway curvature coefficients (f_o), K_{Dmin}, K_{Dmax}, ε, e, and ζ. In addition, this problem has nine constraints, establishing a complex multi-constraint engineering design problem. Tables 21 and 22 compare the optimal solutions and constraints obtained by different algorithms, while the statistical optimization results of the algorithms are given in Table 23. The problem is formulated as follows:

$$\max C_d = \begin{cases} f_c Z^{\frac{2}{3}} D_b^{1.8}, & \text{if} \quad D \leq 25.4\text{mm} \\ 3.647 f_c Z^{\frac{2}{3}} D_b^{1.4}, & \text{if} \quad D > 25.4\text{mm} \end{cases} \tag{20}$$

subject to

$$\begin{aligned}
g_1(x) &= -\frac{\phi_0}{2\sin^{-1}\left(D_b/D_m\right)} + Z - 1 \leqslant 0, \\
g_2(x) &= -2D_b + K_{D\min}(D-d) \leqslant 0, \\
g_3(x) &= -K_{D\max}(D-d) + 2D_b \leqslant 0, \\
g_4(x) &= \zeta B_w - D_b \leq 0, \\
g_5(x) &= -D_m + 0.5(D+d) \leqslant 0, \\
g_6(x) &= -(0.5+e)(D+d) + D_m \leqslant 0, \\
g_7(x) &= -0.5\left(D - D_m - D_b\right) + \varepsilon D_b \leqslant 0, \\
g_8(x) &= 0.515 - f_i \leqslant 0, \\
g_9(x) &= 0.515 - f_i \leqslant 0.
\end{aligned}$$

where

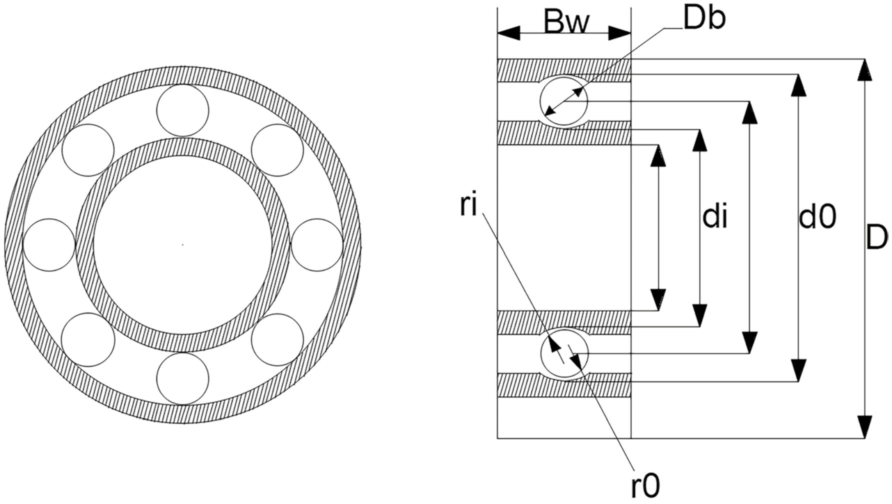

Fig. 21 Schematic of the rolling element-bearing design problem

Table 21 Best solutions obtained from ISSWOA and other algorithms for the rolling element-bearing design problem

Algorithm	Design variable					Optimum cost
	D_m	D_b	Z	f_0	f_i	
ISSWOA	125.6978	21.4233	10.9955	0.515	0.515	87,916.1318
HHO-PS	126.3365	21.03971	11.19893	0.5150	0.5150	85,502.8
SCA	125	21.03287	10.96571	0.515	0.515	83,431.117
HSSAPSO	125.7227	21.4233	11.0012	0.515	0.515	85,539.189
MBA	125.7153	21.423300	11.000	0.515	0.515	85,535.9611
CSA	125	21.418	11.356	0.515	0.515	85,201.641
	$K_{D\min}$	$K_{D\max}$	ε	e	ζ	
ISSWOA	0.4000	0.6895	0.3000	0.0541	0.6342	
HHO-PS	0.4	0.607449	0.3	0.02938	0.6	
SCA	0.5	0.7	0.3	0.027780	0.62912	
HSSAPSO	0.4515	0.6170	0.3000	0.0244	0.6259	
MBA	0.488805	0.627829	0.300149	0.097305	0.646095	
CSA	0.4	0.7	0.3	0.02	0.612	

Table 22 The constraints of ISSWOA and other algorithms for the rolling element-bearing design problem

Constraints	ISSWOA	HHO-PS	SCA	HSSAPSO	MBA	CSA
g_1	−0.0037	−6.33E−06	−0.1269797	4.17E−05	0	0.410909
g_2	−14.8456	−14.07942	−7.06574	−11.2416	−8.630183	−14.836
g_3	−5.4170	−0.44201	−6.93426	−0.3434	−1.101429	−6.164
g_4	−2.3968	−3.03971	−2.15927	−2.6463	−2.040448	−3.058
g_5	−0.6978	−1.3365	0	−0.7227	−0.715366	0
g_6	−12.8378	−6.0085	−6.945	−5.3773	−23.611002	−5
g_7	−0.0115	1.80E−05	−0.673704	−1.00E−05	−0.000480	−0.3656
g_8	0	0	0	0	0	0
g_9	0	0	0	0	0	0

Table 23 Comparison of statistical results using ISSWOA and other algorithm for the rolling element-bearing design problem

Algorithm	Mean	Std	Best	Worst
ISSWOA	87,766.09156	88.85089525	87,916.13187	87,631.39132
HHO-PS	84,536.7	873.9887	85,502.8	83,671.9
SCA	N/A	N/A	83,431.117	N/A
HSSAPSO	N/A	N/A	85,539.189	N/A
MBA	85,321.4030	211.52	85,535.9611	84,440.1948
CSA	85,157.112	117.123	85,201.641	850,945.807

$$f_c = 37.91\left[1+\left\{1.04\left(\frac{1-\gamma}{1+\gamma}\right)^{1.72}\left(\frac{f_i(2f_0-1)}{f_0(2f_i-1)}\right)^{0.41}\right\}^{10/3}\right]^{-0.3}\left[\frac{\gamma^{0.3}(1-\gamma)^{1.39}}{(1+\gamma)^{\frac{1}{3}}}\right]\left[\frac{2f_i}{2f_i-1}\right]^{0.41},$$

$$\phi_0 = 2\pi - 2\cos^{-1}\left(\frac{\left[\left\{\frac{D-d}{2}-3\left(\frac{T}{4}\right)\right\}^2+\left\{\frac{D}{2}-\frac{T}{4}-D_b\right\}^2-\left\{\frac{d}{2}+\frac{T}{4}\right\}^2\right]}{2\left\{\frac{D-d}{2}-3\left(\frac{T}{4}\right)\right\}\left\{\frac{D}{2}-\frac{T}{4}-D_b\right\}}\right),$$

$$\gamma = \frac{D_b}{D_m},\quad f_i = \frac{r_i}{D_b},\quad f_0 = \frac{r_0}{D_b},\quad T = D-d-2D_b,$$

$$D = 160,\quad d = 90,\quad B_w = 30,\quad r_i = r_0 = 11.033.$$

The variable ranges are

$$0.5(D+d) \le D_m \le 0.6(D+d),\quad 0.15(D-d) \le D_m \le 0.45(D-d),$$
$$4 \le Z \le 50,\quad 0.515 \le f_i,\quad f_0 \le 0.6, 0.4 \le K_{D\min} \le 0.5,$$
$$0.6 \le K_{D\max} \le 0.7,\quad 0.3 \le \varepsilon \le 0.4,\quad 0.3 \le e \le 0.4,$$
$$\text{and}\quad 0.6 \le \zeta \le 0.85$$

6.6 Car side impact design problem

The objective of the car side impact design problem [54] is to find the most appropriate combination of variables to minimize the weight of the car door. The design variables are as follows: thicknesses of B-Pillar inner (x_1), thickness of B-Pillar reinforcement (x_2), thickness of floor side inner (x_3), thickness of cross members (x_4), thickness of door beam (x_5), thickness of door beltline reinforcement (x_6), thickness of roof rail (x_7), materials of B-Pillar inner (x_8), materials of floor side inner (x_9), barrier height (x_{10}), and hitting position (x_{11}). Tables 24 and 25 compare the optimal solutions and constraints obtained by different algorithms, while the statistical optimization results of the algorithms are given in Table 26. The problem is formulated as follows:

$$\min f(x) = 1.98 + 4.90x_1 + 6.67x_2 + 6.98x_3 + 4.01x_4 + 1.78x_5 + 2.73x_7 \quad (21)$$

subject to

$$g_1 = 1.16 - 0.3717x2x4 - 0.0931x2x10 - 0.484x3x9 + 0.01343x6x10 - 1 \le 0$$

$$g_2 = 0.261 - 0.0159x_1x_2 - 0.188x_1x_8 - 0.019x_2x_7 + 0.0144x_3x_5 + 0.0008757x_5x_{10} + 0.080405x_6x_9 + 0.00139x_8x_{11} + 0.00001575x_{10}x_{11} - 0.32 \le 0$$

$$g_3 = 0.214 + 0.00817x_5 - 0.131x_1x_8 - 0.0704x_1x_9 + 0.03099x_2x_6 - 0.018x_2x_7 + 0.0208x_3x_8 + 0.121x_3x_9 - 0.00364x_5x_6 + 0.0007715x_5x_{10} - 0.0005354x_6x_{10} + 0.00121x_8x_{11} \le 0.32$$

$$g_4 = 0.074 - 0.061x_2 - 0.163x_3x_8 + 0.001232x_3x_{10} - 0.166x_7x_9 + 0.227x_2^2 \le 0.32$$

$$g_5 = 28.98 + 3.818x_3 - 4.2x_1x_2 + 0.0207x_5x_{10} + 6.63x_6x_9 - 7.7x_7x_8 + 0.32x_9x_{10} - 32 \leqslant 0$$

$$g_6 = 33.86 + 2.95x_3 + 0.1792x_{10} - 5.057x_1x_2 - 11x_2x_8 - 0.0215x_5x_{10} - 9.98x_7x_8 + 22x_8x_9 - 32 \leqslant 0$$

$$g_7 = 46.36 - 9.9x_2 - 12.9x_1x_9 + 0.1107x_3x_{10} - 32 \leqslant 0$$

$$g_8 = 4.72 - 0.5x_4 - 0.19x_2x_3 - 0.0122x_4x_{10} + 0.009325x_6x_{10} - 0.0198x_4x_{10} + 0.028x_6x_{10} - 9.9 \leqslant 0$$

$$g_9 = 10.58 - 0.674x_1x_2 - 1.95x_2x_8 + 0.02054x_3x_{10} - 0.0198x_4x_{10} + 0.028x_6x_{10} - 9.9 \le 0$$

$$g_{10} = 16.45 - 0.489x_3x_7 - 0.843x_5x_6 + 0.0432x_9x_{10} - 0.0556x_9x_{11} - 0.000786x_{11}^2 - 15.7 \le 0$$

The variable ranges are $0.5 \leqslant x_1 \sim x_7 \leqslant 1.5, x_8, x_9 \in [0.192, 0.345], -30 \leqslant x_{10}, x_{11} \leqslant 30$.

Table 24 Best solutions obtained from ISSWOA and other algorithms for the car side impact design problem

Algorithm	Design variable						Optimum cost
	x_1	x_2	x_3	x_4	x_5	x_6	
ISSWOA	0.5000	1.1162	0.5000	1.3027	0.5000	1.5000	22.8375
BSA-SAEε	0.5000	1.1222	0.5000	1.2936	0.5000	1.4997	23.1300
BA	0.5000	1.1167	0.5000	1.3021	0.5000	1.5000	22.8447
GA	0.5000	1.2802	0.5000	1.0330	0.5000	0.5000	22.8565
PSO	0.5000	1.1167	0.5000	1.3021	0.5000	1.5000	22.8447
	x_7	x_8	x_9	x_{10}	x_{11}		
ISSWOA	0.5000	0.3450	0.1920	− 19.5921	0.5194		
BSA-SAEε	0.5000	0.3450	0.1920	− 18.5877	− 0.0375		
BA	0.5000	0.3450	0.1920	− 19.5494	− 0.0043		
GA	0.5000	0.3499	0.1920	10.3119	0.0017		
PSO	0.5000	0.3450	0.1920	− 19.5494	− 0.0043		

Table 25 The constraint of ISSWOA and other algorithms in car side impact design problem

Constraint	BSA-SAEε	BA	GA	PSO	ISSWOA
g_1	− 0.606238976	− 0.61750525	0.43167	− 0.617505	− 0.618005259
g_2	− 0.101289326	− 0.09270723	0.09839	− 0.092707	− 0.102273912
g_3	− 0.10075536	− 0.10062705	0.13685	− 0.100627	− 0.100402609
g_4	− 0.034180971	− 0.03420961	0.02699	− 0.034209	− 0.034185453
g_5	− 4.221291988	− 4.27832783	3.77008	− 4.278328	− 4.280298281
g_6	− 7.156847652	− 7.28381045	3.36103	− 7.2838105	− 7.287758914
g_7	− 0.004253534	− 0.00263652	0.00025	− 0.0026365	− 2.22E−07
g_8	− 5.45E−06	− 2.43E−05	8.6156E−06	− 2.425E−05	− 2.57E−07
g_9	− 0.948520903	− 0.96542696	0.58567	− 0.9654269	− 0.965839618
g_{10}	− 0.158152439	− 0.16660413	0.502506	− 0.1666041	− 0.172773903

Table 26 Comparison of statistical results using ISSWOA and other algorithm for the car side impact design problem

Algorithm	Mean	Std	Best	Worst
ISSWOA	22.8692877127973	0.0165021626	22.843987015	22.89555
BSA-SAEε	22.96220	0.07659	22.96220	23.13001
BA	22.89273	0.17383	22.84474	23.21354
GA	23.51585	0.66555	22.8565	26.240578
PSO	22.89429	0.15017	22.84474	23.21354

6.7 Gear train design problem

This gear train design problem [52] is shown in Fig. 22. It is an unconstrained design problem. It is necessary to find the best gear number combination. There are four gears in this problem, with number of teeth $M_A(x_1)$, $M_B(x_2)$, $M_C(x_3)$, and $M_D(x_4)$. Table 27 compares the optimal solutions obtained by different algorithms, while the statistical optimization results of the algorithms are given in Table 28. The equation for the gear train design problem is shown below

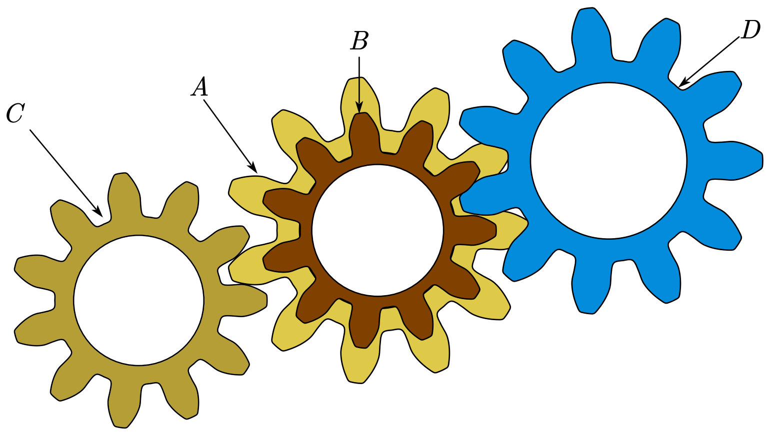

Fig. 22 Schematic of the gear train

Table 27 Best solutions obtained from ISSWOA and other algorithms for the gear train

Algorithm	Design variable				Optimum cost
	M_A	M_B	M_C	M_D	
ISSWOA	49	8	38	43	2.700857E−12
MBA	43	16	19	49	2.700857E−12
SCA-GWO	51	26	15	53	2.3078E−11
OBSCA	43	12	31	60	8.7008E−09
PSO	47	13	12	23	9.9216E−10

Table 28 Comparison of statistical results using ISSWOA and other algorithm for the gear train

Algorithm	Mean	Std	Best	Worst
ISSWOA	1.848907E−11	7.854368E−11	2.700857E−12	4.334527E−10
MBA	2.471635E−09	3.94E−09	2.700857E−12	2.062904E−08
SCA-GWO	N/A	N/A	2.3078E−11	N/A
OBSCA	N/A	N/A	8.7008E−09	N/A
PSO	N/A	N/A	9.9216E−10	N/A

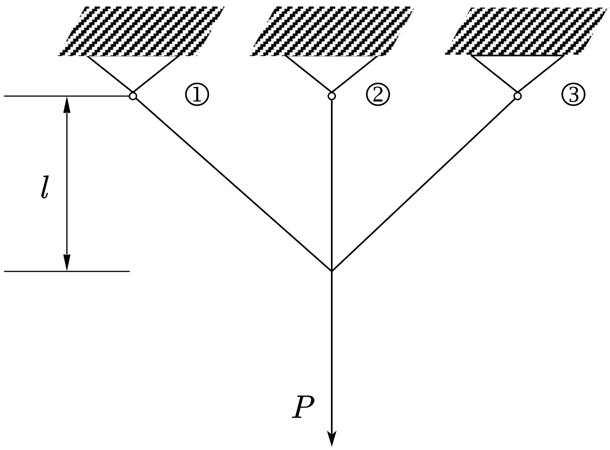

Fig. 23 Schematic of three-bar truss

Table 29 Best solutions obtained from ISSWOA and other algorithms for the three-bar truss

Algorithm	Design variable		Optimum cost
	A1	*A2*	
ISSWOA	0.788685663	0.408218514	263.89584
MFO	0.78824477	0.409466905	263.895979
MBA	0.7885650	0.4085597	263.895852
IGWO	0.78846	0.40884	263.8959

Table 30 The constraint of ISSWOA and other algorithms in car side impact design problem

Constraint	ISSWOA	MFO	MBA	IGWO
g_1	−1.17356E−09	2.5919661E−09	1.41887E−07	1.29653208E−05
g_2	−1.464135466	−1.46271707206	−1.463747582	−1.463422650022
g_3	−0.535864534	−0.5372829253	−0.536252276	−0.536564384656

$$\min f(x_1, x_2, x_3, x_4) = \left(\frac{1}{6.931} - \frac{x_3 x_2}{x_1 x_4} \right) \tag{22}$$

The variable ranges are $12 \leqslant x_1, x_2, x_3, x_4 \leqslant 60$.

Table 31 Comparison of statistical results using ISSWOA and other algorithms for the three-bar truss

Algorithm	Mean	Std	Best	Worst
ISSWOA	263.8977624355	0.002715958982	263.89584	263.906647
MFO	NA	N/A	263.895979	N/A
MBA	NA	N/A	263.895852	N/A
IGWO	NA	N/A	263.8959	N/A

6.8 Three-bar truss design problem

The structure of the three-bar truss [52] is shown in Fig. 23. To minimize the volume of the three-bar truss and meet the stress constraints on each side of the truss member, the most appropriate cross-section combination of the truss member is found. Tables 29 and 30 show the comparison among the optimal solutions and constraints obtained by different algorithms, and Table 31 gives the statistical optimization results of the algorithms. The equation is as follows:

$$\min f(x) = [A1, A2] = \left(2\sqrt{2}x_1 + x_2\right) \times l \tag{23}$$

$$\begin{aligned} g_1(x) &= \frac{\sqrt{2}x_1 + x_2}{\sqrt{2}x_1^2 + 2x_1x_2} P - \sigma \leqslant 0 \\ g_2(x) &= \frac{x_2}{\sqrt{2}x_1^2 + 2x_1x_2} P - \sigma \leqslant 0 \text{ where } l = 100\,\text{cm}, P = 2\frac{\text{KN}}{\text{cm}^2}, \sigma = 2\frac{\text{KN}}{\text{cm}^2}. \\ g_3(x) &= \frac{1}{\sqrt{2}x_2 + x_1} P - \sigma \leqslant 0 \end{aligned}$$

The variable ranges are $0 \leqslant x_1, x_2 \leqslant 1$.

6.9 Economic load dispatch

Economic load dispatch (ELD) [59] is an important engineering problem in power delivery systems. The goal of this problem is to find the best power distribution of available thermal units, to minimize the fuel cost while meeting the load. The calculation formula of the fuel cost is as follows:

$$F(Pg) = \sum_{i=1}^{n} \left(a_i Pg_i^2 + b_i Pg_i + c_i\right) + \left| d_i \sin\left(e_i\left(Pg_i^{\min} - Pg_i\right)\right)\right| \tag{24}$$

where a_i, b_i, and c_i are the fuel-cost coefficients of the ith unit, and d_i and e_i are the fuel-cost coefficients of the ith unit with valve-point effects.

Table 32 The best results obtained by the ISSWOA for ELD

Unit	ISSWOA	Unit	ISSWOA	Unit	ISSWOA	Unit	ISSWOA
Pg1	102.3826	Pg11	229.0685	Pg21	520.5743	Pg31	156.4526
Pg2	103.8697	Pg12	236.4263	Pg22	519.7126	Pg32	165.7370
Pg3	114.3121	Pg13	298.7664	Pg23	524.0416	Pg33	154.1284
Pg4	180.4400	Pg14	303.9936	Pg24	523.3120	Pg34	190.9589
Pg5	91.5125	Pg15	300.1537	Pg25	520.3682	Pg35	188.4636
Pg6	135.1597	Pg16	304.6267	Pg26	523.1161	Pg36	187.2088
Pg7	282.7740	Pg17	475.6833	Pg27	10.0037	Pg37	62.2194
Pg8	285.8164	Pg18	481.2101	Pg28	11.4143	Pg38	51.8958
Pg9	281.2206	Pg19	517.0165	Pg29	12.9032	Pg39	48.3053
Pg10	279.8587	Pg20	511.6479	Pg30	91.6154	Pg40	521.5098
Total power output (MW)		10,500		Total fuel cost ($/h)		117,892.5544	

Table 33 Comparison of statistical results using ISSWOA and other algorithm for the car side impact design problem

Algorithm	Mean	Std	Best	Worst
ISSWOA	119,271.3761	1090.919397	117,892.5544	121,156.837
GJO	122,146.9161	N/A	121,586.8532	123,095.462
HPSOM	124,350.87	N/A	122,112.40	N/A
HPSOWM	122,844.40	N/A	121,915.30	N/A
HGPSO	126,855.70	N/A	124,797.13	N/A

In this research, the number of units was set as 40, and the load was 10,500 MW. Loss of thermal units was considered negligible. The upper and lower limits of each unit of power and the value of the coefficients in Eq. 24 are given in Sinha et al. [66]. The best result obtained by ISSWOA is shown in Table 32. The comparison with other algorithm results is shown in Table 33.

It can be seen from the test results of the eight engineering problems that ISSWOA performed best on the element-bearing design problem and ELD problem, clearly outperforming other algorithms in terms of stability and optimal results. For the other six engineering problems, ISSWOA also performed adequately, which showed that ISSWOA is certainly viable for solving practical engineering problems.

7 Conclusion

This study proposed a hybrid ISSWOA that combines the improved SSA with Levy flight strategy and the improved WOA with a novel spiral updating strategy. The performance of ISSWOA was investigated on standard unimodal, multimodal, and fixed-dimensional multimodal benchmark functions. The purpose was to determine its capabilities on three primary criteria: avoidance of trapping in local optima and exploration and exploitation abilities. The results showed that ISSWOA performed significantly better than other metaheuristic algorithms on most benchmark functions.

Since metaheuristic algorithms are suited to complex engineering problems, this paper applied ISSWOA to seven kinds of engineering design problems and a large electrical engineering problem (i.e., the ELD problem). The results showed that ISSWOA achieved good performance, especially in the element-bearing design problem and ELD problem, and was effective compared to other algorithms in other engineering problems.

In this research, the parameter configuration of ISSWOA was balanced to suit different problems However, different parameter configurations will be better suited to certain problems; therefore, the next research objective is to adapt the parameter configuration to different problems. In addition, the binary and multi-objective versions of ISSWOA can be developed further to solve a wider variety of problems.

Author contributions JZ and XC proposed the innovation and designed the experiment in this study, JZ, MZ, and JL performed the simulation experiments and analyzed the experiment results and wrote the manuscript, JL corrected the manuscript.

Funding This work was supported by the National Natural Science Foundation of China Program under Grant 62073198.

Data availability All data generated or analyzed during this study are included in this article.

Declarations

Conflict of interest The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Ethical approval This study will not cause harm to anyone or animals.

References

1. Zhang Y, Mo Y (2022) Chaotic adaptive sailfish optimizer with genetic characteristics for global optimization. J Supercomput 78:10950–10996. https://doi.org/10.1007/s11227-021-04255-9
2. Tang KS, Man KF, Kwong S et al (2022) Genetic algorithms and their applications. IEEE Signal Proc Mag 13:22–37. https://doi.org/10.1109/79.543973
3. Das S, Suganthan PN (2010) Differential evolution: a survey of the state-of-the-art. IEEE Trans Evol Comput 15:4–31. https://doi.org/10.1109/TEVC.2010.2059031
4. Lee KY, Yang FF (1998) Optimal reactive power planning using evolutionary algorithms: a comparative study for evolutionary programming, evolutionary strategy, genetic algorithm, and linear programming. IEEE Trans Power Syst 13:101–108. https://doi.org/10.1109/59.651620
5. Espejo PG, Ventura S, Herrera F (2009) A survey on the application of genetic programming to classification. IEEE Trans Syst Man Cybernet C 40:121–144. https://doi.org/10.1109/TSMCC.2009.2033566
6. Zhong J, Feng L, Ong YS (2017) Gene expression programming: a survey. IEEE Comput Intell Mag 12:54–72

7. Prajapati A (2022) A customized PSO model for large-scale many-objective software package restructuring problem. Arab J Sci Eng 47:10147–10162. https://doi.org/10.1007/s13369-021-06523-5
8. Mirjalili S, Mirjalili SM, Lewis A (2014) Grey wolf optimizer. Adv Eng Soft 69:46–61. https://doi.org/10.1016/j.advengsoft.2013.12.007
9. Hashim FA, Hussien AG (2022) Snake optimizer: a novel meta-heuristic optimization algorithm. Knowl Based Syst 242:108320. https://doi.org/10.1016/j.knosys.2022.108320
10. Salgotra R, Singh U, Saha S (2019) On some improved versions of whale optimization algorithm. Arab J Sci Eng 44:9653–9691. https://doi.org/10.1007/s13369-019-04016-0
11. Peraza-Vázquez H, Peña-Delgado AF, Echavarría-Castillo G et al (2021) A bio-inspired method for engineering design optimization inspired by dingoes hunting strategies. Math Probl Eng. https://doi.org/10.1155/2021/9107547
12. Zhang Z, He R, Yang K (2022) A bioinspired path planning approach for mobile robots based on improved sparrow search algorithm. Adv Manuf 10:114–130. https://doi.org/10.1007/s40436-021-00366-x
13. Gürses D, Mehta P, Sait SM et al (2022) African vultures optimization algorithm for optimization of shell and tube heat exchangers. Mater Test 64:1234–1241. https://doi.org/10.1515/mt-2022-0050
14. Barbarosoglu G, Ozgur D (1999) A tabu search algorithm for the vehicle routing problem. Comput Oper Res 26:255–270. https://doi.org/10.1016/S0305-0548(98)00047-1
15. Dai C, Chen W, Zhu Y et al (2009) Seeker optimization algorithm for optimal reactive power dispatch. IEEE Trans Power Syst 24:1218–1231. https://doi.org/10.1109/TPWRS.2009.2021226
16. Ramezani F, Lotfi S (2013) Social-based algorithm (SBA). Appl Soft Comput 13:2837–2856. https://doi.org/10.1016/j.asoc.2012.05.018
17. Ghorbani N, Babaei E (2014) Exchange market algorithm. Appl Soft Comput 19:177–187. https://doi.org/10.1016/j.asoc.2014.02.006
18. Vincent FY, Jewpanya P, Redi AANP et al (2021) Adaptive neighborhood simulated annealing for the heterogeneous fleet vehicle routing problem with multiple cross-docks. Comput Oper Res 129:105205. https://doi.org/10.1016/j.cor.2020.105205
19. Pashaei E, Aydin N (2017) Binary black hole algorithm for feature selection and classification on biological data. Appl Soft Comput 56:94–106. https://doi.org/10.1016/j.asoc.2017.03.002
20. Eskandar H, Sadollah A, Bahreininejad A et al (2012) Water cycle algorithm—a novel metaheuristic optimization method for solving constrained engineering optimization problems. Comput Struct 110:151–166. https://doi.org/10.1016/j.compstruc.2012.07.010
21. Kaveh A, Khayatazad M (2012) A new meta-heuristic method: ray optimization. Comput Struct 112:283–294. https://doi.org/10.1016/j.compstruc.2012.09.003
22. Kumar S, Tejani GG, Pholdee N et al (2021) Hybrid heat transfer search and passing vehicle search optimizer for multi-objective structural optimization. Knowl Based Syst 212:106556. https://doi.org/10.1016/j.knosys.2020.106556
23. Yildiz AR, Mehta P (2022) Manta ray foraging optimization algorithm and hybrid Taguchi salp swarm-Nelder–Mead algorithm for the structural design of engineering components. Mater Test 64:706–713. https://doi.org/10.1515/mt-2022-0012
24. Li Q, Wang W (2021) AVO inversion in orthotropic media based on SA-PSO. IEEE Trans Geosci Remote 99:1–10. https://doi.org/10.1109/TGRS.2021.3053044
25. Mafarja MM, Mirjalili S (2017) Hybrid whale optimization algorithm with simulated annealing for feature selection. Neurocomputing 260:302–312. https://doi.org/10.1016/j.neucom.2017.04.053
26. Laskar NM, Guha K, Chatterjee I et al (2019) HWPSO: a new hybrid whale-particle swarm optimization algorithm and its application in electronic design optimization problems. Appl Intell 49:265–291. https://doi.org/10.1007/s10489-018-1247-6
27. Han X, Yue L, Dong Y et al (2020) Efficient hybrid algorithm based on moth search and fireworks algorithm for solving numerical and constrained engineering optimization problems. J Supercomput 76:9404–9429. https://doi.org/10.1007/s11227-020-03212-2
28. Shehab M, Khader AT, Laouchedi M et al (2019) Hybridizing cuckoo search algorithm with bat algorithm for global numerical optimization. J Supercomput 75:2395–2422. https://doi.org/10.1007/s11227-018-2625-x
29. Li X, Gu J, Sun X et al (2022) Parameter identification of robot manipulators with unknown payloads using an improved chaotic sparrow search algorithm. Appl Intell. https://doi.org/10.1007/s10489-021-02972-5

30. Chakraborty S, Sharma S, Saha AK et al (2021) SHADE–WOA: a metaheuristic algorithm for global optimization. Appl Soft Comput 113:107866. https://doi.org/10.1016/j.asoc.2021.107866
31. Xue J, Shen B (2020) A novel swarm intelligence optimization approach: sparrow search algorithm. Syst Sci Control Eng 8:22–34. https://doi.org/10.1080/21642583.2019.1708830
32. Mirjalili S, Lewis A (2016) The whale optimization algorithm. Adv Eng Softw 95:51–67. https://doi.org/10.1016/j.advengsoft.2016.01.008
33. Iacca G, dos Santos JVC, de Melo VV (2021) An improved Jaya optimization algorithm with Lévy flight. Expert Syst Appl 165:113902. https://doi.org/10.1016/j.eswa.2020.113902
34. Alsattar HA, Zaidan AA, Zaidan BB (2020) Novel meta-heuristic bald eagle search optimisation algorithm. Artif Intell Rev 53:2237–2264. https://doi.org/10.1007/s10462-019-09732-5
35. Nadimi-Shahraki MH, Taghian S, Mirjalili S (2021) An improved grey wolf optimizer for solving engineering problems. Expert Syst Appl 166:113917. https://doi.org/10.1016/j.eswa.2020.113917
36. Zhang M, Long D, Qin T et al (2020) A chaotic hybrid butterfly optimization algorithm with particle swarm optimization for high-dimensional optimization problems. Symmetry 12:1800. https://doi.org/10.3390/sym12111800
37. Khalilpourazari S, Khalilpourazary S (2019) An efficient hybrid algorithm based on water cycle and moth-flame optimization algorithms for solving numerical and constrained engineering optimization problems. Soft Comput 23:1699–1722. https://doi.org/10.1007/s00500-017-2894-y
38. Tang A, Zhou H, Han T et al (2021) A chaos sparrow search algorithm with logarithmic spiral and adaptive step for engineering problems. CMES Compt Model Eng 130:331–364. https://doi.org/10.32604/cmes.2021.017310
39. Mirjalili S (2015) The ant lion optimizer. Adv V Eng Softw 83:80–98. https://doi.org/10.1016/j.advengsoft.2015.01.010
40. Dhiman G, Kumar V (2017) Spotted hyena optimizer: a novel bio-inspired based metaheuristic technique for engineering applications. Adv Eng Softw 114:48–70. https://doi.org/10.1016/j.advengsoft.2017.05.014
41. Krishna AB, Saxena S, Kamboj VK (2021) A novel statistical approach to numerical and multidisciplinary design optimization problems using pattern search inspired Harris hawks optimizer. Neural Comput Appl 33:7031–7072. https://doi.org/10.1007/s00521-020-05475-5
42. Kamboj VK, Nandi A, Bhadoria A et al (2020) An intensify Harris Hawks optimizer for numerical and engineering optimization problems. Appl Soft Comput 89:106018. https://doi.org/10.1016/j.asoc.2019.106018
43. Ferreira MP, Rocha ML, Neto AJS et al (2018) A constrained ITGO heuristic applied to engineering optimization. Expert Syst Appl 110:106–124. https://doi.org/10.1016/j.eswa.2018.05.027
44. Zhang Z, Ding S, Jia W (2019) A hybrid optimization algorithm based on cuckoo search and differential evolution for solving constrained engineering problems. Eng Appl Artif Intel 85:254–268. https://doi.org/10.1016/j.engappai.2019.06.017
45. Zahara E, Kao YT (2009) Hybrid Nelder–Mead simplex search and particle swarm optimization for constrained engineering design problems. Expert Syst Appl 36:3880–3886. https://doi.org/10.1016/j.eswa.2008.02.039
46. Guo W, Chen M, Wang L et al (2016) Backtracking biogeography-based optimization for numerical optimization and mechanical design problems. Appl Intell 44:894–903. https://doi.org/10.1007/s10489-015-0732-4
47. Baykasoğlu A, Ozsoydan FB (2015) Adaptive firefly algorithm with chaos for mechanical design optimization problems. Appl Soft Comput 36:152–164. https://doi.org/10.1016/j.asoc.2015.06.056
48. Savsani P, Savsani V (2016) Passing vehicle search (PVS): a novel metaheuristic algorithm. Appl Math Model 40:3951–3978. https://doi.org/10.1016/j.apm.2015.10.040
49. Hashim FA, Houssein EH, Mabrouk MS et al (2019) Henry gas solubility optimization: a novel physics-based algorithm. Future Gener Comput Syst 101:646–667. https://doi.org/10.1016/j.future.2019.07.015
50. Abd EM, Oliva D, Xiong S (2017) An improved opposition-based sine cosine algorithm for global optimization. Expert Syst Appl 90:484–500. https://doi.org/10.1016/j.eswa.2017.07.043
51. Singh N, Singh SB, Houssein EH (2020) Hybridizing salp swarm algorithm with particle swarm optimization algorithm for recent optimization functions. Evol Intell 1:1–34. https://doi.org/10.1007/s12065-020-00486-6

52. Sadollah A, Bahreininejad A, Eskandar H et al (2013) Mine blast algorithm: a new population based algorithm for solving constrained engineering optimization problems. Appl Soft Comput 13:2592–2612. https://doi.org/10.1016/j.asoc.2012.11.026
53. Braik MS (2021) Chameleon swarm algorithm: a bio-inspired optimizer for solving engineering design problems. Expert Syst Appl 174:114685. https://doi.org/10.1016/j.eswa.2021.114685
54. Zhang C, Lin Q, Gao L et al (2015) Backtracking search algorithm with three constraint handling methods for constrained optimization problems. Expert Syst Appl 42:7831–7845. https://doi.org/10.1016/j.eswa.2015.05.050
55. Yang XS, Gandomi AH (2012) Bat algorithm: a novel approach for global engineering optimization. Eng Comput. https://doi.org/10.1108/02644401211235834
56. Singh N, Singh SB (2017) A novel hybrid GWO-SCA approach for optimization problems. Eng Sci Technok 20:1586–1601. https://doi.org/10.1016/j.jestch.2017.11.001
57. Mirjalili S (2015) Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm. Knowl Based Syst 89:228–249. https://doi.org/10.1016/j.knosys.2015.07.006
58. Li Y, Lin X, Liu J (2021) An improved gray wolf optimization algorithm to solve engineering problems. Sustainability 13:3208. https://doi.org/10.3390/su13063208
59. Chopra N, Ansari MM (2022) Golden jackal optimization: a novel nature-inspired optimizer for engineering applications. Expert Syst Appl 198:116924. https://doi.org/10.1016/j.eswa.2022.116924
60. Ling SH, Iu HHC, Chan KY et al (2008) Hybrid particle swarm optimization with wavelet mutation and its industrial applications. IEEE Trans Syst Man Cybernet B 38:743–763. https://doi.org/10.1109/TSMCB.2008.921005
61. Bayzidi H, Talatahari S, Saraee M et al (2021) Social network search for solving engineering optimization problems. Comput Intel Neurosci. https://doi.org/10.1155/2021/8548639
62. Chen H, Wang M, Zhao X (2020) A multi-strategy enhanced sine cosine algorithm for global optimization and constrained practical engineering problems. Appl Math Comput 369:124872. https://doi.org/10.1016/j.amc.2019.124872
63. Chauhan S, Vashishtha G, Kumar A (2022) A symbiosis of arithmetic optimizer with slime mould algorithm for improving global optimization and conventional design problem. J Supercomput 78:6234–6274. https://doi.org/10.1007/s11227-021-04105-8
64. Migallón H, Jimeno-Morenilla A, Rico H et al (2021) Multi-level parallel chaotic Jaya optimization algorithms for solving constrained engineering design problems. J Supercomput 77:12280–12319. https://doi.org/10.1007/s11227-021-03737-0
65. Emami H (2022) Stock exchange trading optimization algorithm: a human-inspired method for global optimization. J Supercomput 78:2125–2174. https://doi.org/10.1007/s11227-021-03943-w
66. Sinha N, Chakrabarti R, Chattopadhyay PK (2003) Evolutionary programming techniques for economic load dispatch. IEEE Trans Evolut Comput 7:83–94. https://doi.org/10.1109/TEVC.2002.806788