

An improved sparrow search algorithm and its application in the deployment of 3D wireless sensor nodes

Received: 21 March 2025

Accepted: 8 December 2025

Published online: 15 December 2025

Cite this article as: Li S., Mo L.P., Yin B. *et al.* An improved sparrow search algorithm and its application in the deployment of 3D wireless sensor nodes. *Sci Rep* (2025). <https://doi.org/10.1038/s41598-025-32080-0>

Shan Li, Li Ping Mo, Biao Yin & Wei Min

We are providing an unedited version of this manuscript to give early access to its findings. Before final publication, the manuscript will undergo further editing. Please note there may be errors present which affect the content, and all legal disclaimers apply.

If this paper is publishing under a Transparent Peer Review model then Peer Review reports will publish with the final article.

An Improved Sparrow Search Algorithm and Its Application in the Deployment of 3D Wireless Sensor Nodes

Shan Li ², Li Ping Mo ¹✉, Biao Yin ², Wei Min ²

¹College of Computer Science and Engineering, Jishou University, Jishou 416000, China.

✉ e-mail: zmx89@jsu.edu.cn.

²School of Communication and Electronic Engineering, Jishou University, Jishou 416000, China.

e-mail: shanli@stu.jsu.edu.cn; yb49@stu.jsu.edu.cn; minwei@stu.jsu.edu.cn

This manuscript proposes an Improved Sparrow Search Algorithm with Multi-Strategies (MSISSA) to address the shortcomings of the Sparrow Search Algorithm (SSA), and applies it to optimize the deployment of 3D Wireless Sensor Nodes (3D-WSNs). The improvements of MSISSA encompass three main components: firstly, it utilizes an adaptive weight operator for updating position to enhance local search accuracy and expedite convergence; secondly, it introduces the spiral flight mechanism of the Moth Flame Optimization (MFO) algorithm, which balances the algorithm's local and global search abilities; finally, it uses the Levy flight mechanism to jump out the local optima. The experimental results demonstrate that MSISSA outperforms SSA, its four variants, and two classical intelligent algorithms. The simulation results indicate that when the number of nodes is 30 and 50, the optimal coverage rate of MSISSA reaches 91.89% and 99.11%, respectively, surpassing those achieved by SSA and its four variants, as well as two classical intelligent algorithms. Therefore, it can be concluded that MSISSA is well-suited for optimizing the deployment of 3D-WSNs.

Keywords Sparrow search algorithm, Moth-flame optimization algorithm, Levy flight, Three-dimensional wireless sensor nodes.

1 Introduction

Metaheuristic algorithms, which are a class of algorithms designed to tackle complex optimization problems, progressively approximate the optimal solution by searching within the problem space without requiring detailed assumptions about the problem's underlying model. In recent years, metaheuristic algorithms represented by Particle Swarm Optimization (PSO) Algorithm [1], Genetic Algorithm (GA) [2], Cuckoo Search Algorithm (CS) [3], Moth-Flame Optimization (MFO) Algorithm [4], Grasshopper Optimization Algorithm (GOA) [5], Grey Wolf Optimizer (GWO) Algorithm [6], and Egret Swarm Optimization Algorithm (ESOA) [7], have been continuously improved and widely applied to solve complex engineering problems, such as path planning [8-10] and network optimizing [11-13]. Wu et al. [14] proposed a multi-attribute label-based ant colony system (LACS) algorithm to address vehicle heterogeneity and mixed backhaul loads. Liu et al. [15] presented a new three-stage hybrid feature gene selection method. Huang et al. [16] utilized a nonlinear activated beetle antennae search algorithm for fraud detection of public trading companies. Ye et al. [17] explored a multi-objective Cuckoo Search Algorithm to optimize coordinated formation control for multiple drones, ensuring safe flight paths while aligning airspeed, yaw angle, altitude, and altitude rate within specified parameters. Ou et al. [18] introduced an enhanced multi-strategy GWO algorithm for wireless sensor networks. Kang et al. [19] proposed a harmony search algorithm with segmented search capabilities. Khan et al. [20] implemented a human-guided cooperative robot agent in smart homes based on the beetle antennae search. Shang

et al. [21] focused on multi-objective optimization for magnetic resistance magnetic gears using ESOA. S Sharma et al. [22] solve multiple engineering design problems and two multi-gravity-assist spacecraft trajectory problems using an improved butterfly optimization algorithm. SK Sahoo et al. [23] segmentation of COVID-19 CT images using improved moth-flame optimization.

The Sparrow Search Algorithm (SSA) [24], proposed in 2020, is inspired by the foraging and anti-predation behaviors of sparrows. As a novel metaheuristic algorithm, SSA is renowned for its ability to achieve higher optimization accuracy than classical algorithms, and can be applied to solve a variety of engineering problems. However, SSA still faces challenges in solving high-dimensional problems, including a tendency to fall into local optima, slow convergence speed, and poor global search capability. Researchers have proposed some variants of SSA to address these shortcomings. Li et al. [25] introduced an adaptive T-distribution to enhance local search (T_SSA). Fu et al. [26] developed a multi-strategy SSA that combines chaotic reverse learning and Cauchy-Gaussian mutation to balance search capabilities (LSSA). Huang et al. [27] utilized t-distribution and Tent chaotic mapping to increase the diversity of the initial population (T_TentSSA). Zhang et al. [28] combined circular chaotic mapping with the butterfly optimization algorithm and dimension mutation to enhance global exploration (CASSA). The aforementioned variants demonstrate higher performance compared to the standard SSA, but still have various limitations in solving high-dimensional problems, such as slow convergence accuracy and convergence speed, poor global search capability, and susceptibility to local optima.

Aim to solve high-dimensional problems, this manuscript presents an improved Sparrow Search Algorithm with multi-strategy (MSISSA) from enhancing the convergence speed, balancing the ability of local and global search, and escaping the local optima. MSISSA integrates three strategies: adaptive weight operator, the MFO algorithm, and the Levy flight mechanism. To verify performance, ablation experiments are conducted to evaluate the individual contributions of MSISSA components. Subsequently, MSISSA is benchmarked against other seven algorithms: SSA [24], T_SSA [25], LSSA [26], T_TentSSA[27], CASSA [28], PSO [1], and MFO [4] on ten benchmark functions. The results indicate that MSISSA outperforms these seven algorithms. Ultimately, MSISSA is implemented to optimize the deployment of Three-Dimensional Wireless Sensor Nodes (3D-WSNs). Simulation experiments with 30 and 50 nodes demonstrate MSISSA's superior stability compared to its competitors.

The remainder of this manuscript is organized as follows: Section 2 introduces SSA; Section 3 explains the improvements and underlying rationale of the MSISSA; Section 4 presents the analytical experiments; Section 5 discusses the application of MSISSA in the deployment of 3D-WSNs; and Section 6 provides the conclusion.

2 SSA

SSA simulates the foraging and anti-predation behaviors of sparrows. Based on differences in individual foraging behaviors, the sparrow population is divided into discoverers, followers, and sentinels. The discoverers are responsible for searching for food sources, while the followers follow the discoverers to forage. Additionally, a

portion of the sparrows are randomly selected from the population to serve as the sentinels, who notify the group to adopt anti-predation behaviors when danger is detected.

In SSA, 10-20% of the sparrows possessing the optimal fitness values are chosen to serve as discoverers. These discoverers prioritize food acquisition during the search process and are responsible for finding food. The position update formula for the discoverers is as follows (1):

$$X_{i,j}^{t+1} = \begin{cases} X_{i,j}^t \times \exp\left(\frac{\alpha}{iter_{max}}\right) \times \frac{i}{\theta} & \text{if } R_1 < ST \\ X_{i,j}^t + Q \times L & \text{if } R_1 \geq ST \end{cases} \quad j=1,2,\dots,d \quad (1)$$

where $X_{i,j}$ represents the position of the i -th sparrow in the j -th dimension. α is a random number in (0, 1). R_1 and ST denote the warning value in (0, 1) and safety value in (0.5, 1), respectively. Q is a random number drawn from a normal distribution. L is a $1 \times d$ matrix where each element equals 1. t and $iter_{max}$ represent the current iteration number and the maximum number of iterations, respectively. When $R_1 < ST$, it indicates an absence of predators in the foraging environment, allowing discoverers to search more broadly. Conversely, when $R_1 \geq ST$, it signifies that some sparrows have detected predators and alerted others, prompting all sparrows to quickly relocate to safer areas for foraging.

The position update formula for followers is defined by (2).

$$X_{i,j}^{t+1} = \begin{cases} Q \times \exp\left(\frac{\alpha(X_{worst} - X_{i,j}^t)}{i^2}\right) \times \frac{i}{\theta} & \text{if } i > \frac{n}{2} \\ X_p^{t+1} + |X_{i,j}^t - X_p^{t+1}| \times A^+ \times L & \text{otherwise} \end{cases} \quad (2)$$

where X_{worst} and X_p represent the current global worst position and the current optimal position of discoverers, respectively. A is a vector whose elements are randomly assigned values of -1 or 1, and $A^+ = A^T(AA^T)^{-1}$. When $i > n/2$, it indicates that the i -th sparrow, with a lower fitness value, has not obtained food and is in a state of extreme hunger. In this scenario, it must search for energy by foraging in other locations.

In SSA, 10-20% of the sparrows are randomly selected as sentinels, whose task is to detect danger and alert the group to move to a safe foraging area. The position update formula is defined by (3).

$$X_{i,j}^{t+1} = \begin{cases} X_{best}^t + b \times |X_{i,j}^t - X_{best}^t| & \text{if } f_i > f_g \\ X_{i,j}^t + K \times \frac{|X_{i,j}^t - X_{worst}^t|}{\frac{f_i - f_w}{f_i - f_g} + e} & \text{if } f_i = f_g \end{cases} \quad (3)$$

represents current global optimal position. b is a random number that follows a normal distribution. K is a random variable in (-1,1). f_i , f_w and f_g represent the fitness values of the current individual, the current global worst, and the current global optima, respectively. e is an infinitesimal constant to prevent division by zero.

3 The Proposed MSISSA

MSISSA employs an adaptive weight operator to dynamically regulate convergence and facilitate automatic search tuning, integrates the spiral flight mechanism to enhance the balance between global and local search capabilities, and the incorporates the Levy flight mechanism to strengthen the algorithm's ability to escape local optima.

3.1 Position update using an adaptive weight operator for the discoverers

In SSA, when $R_1 < ST$, the discoverer leads followers in foraging. However, as the number of iterations increases, the discoverer's search capability decreases, causing the algorithm to slow down its convergence speed. Meanwhile, when $R_1 \geq ST$, the discoverer at this point will lead the followers to leave their original search position, causing the algorithm to struggle with convergence. To overcome it, MSISSA introduces an adaptive weight operator w with exponential decay [29-31] to the position update formula for the discoverers. Then, formula (1) is replaced with the following formula (4).

$$X_{i,j}^{t+1} = \begin{cases} X_{i,j}^t \times \exp\left(-\frac{w \times iter}{iter_{max}}\right) & \text{if } R_1 < ST \\ X_{i,j}^t + Q \times w & \text{if } R_1 \geq ST \end{cases} \quad (4)$$

where w is defined by (5).

$$w = w_{min} \times \exp\left(-\frac{w_{max} - w_{min}}{iter_{max}} \times iter\right) \quad (5)$$

where w_{min} and w_{max} with an empirical value in (0.1, 0.7), and represent the minimum and maximum values of w , respectively. At the beginning of the iterations, the large value of w ensures the algorithm's global exploration, and as w decreases, the algorithm's convergence speed is enhanced.

3.2 Position update integrating the spiral flight mechanism for the followers

During the process of converging towards a discoverer, followers tend to rapidly gather around the discoverer, which may lead the algorithm to get trapped in a local optimum, and weaken its global search capability. To overcome this issue, MSISSA integrates the spiral flight mechanism of the MFO algorithm into the position update formula for the followers.

MFO algorithm is an innovative optimization algorithm that mimics the flight mechanism of moths. In MFO, a population is first initialized within the feasible domain, and their corresponding fitness values are calculated. The moths' spatial positions are then sorted in ascending order based on their fitness values, and these positions are assigned as the locations of the flames. The position update for the moths is calculated using Equation (6).

$$M_i = S(M_i, F_j) \quad (6)$$

where M_i represents the position of the i -th moth, F_j represent the position of the j -th flame, and S is a spiral function defined by (7).

$$S(M_i, F_j) = D_i \times \exp(b \times r) \times \cos(2\pi \times r) + F_j \quad (7)$$

where D_i defined by (8), represents the distance between the i -th moth and the j -th flame. b is a constant that defines the shape of the logarithmic spiral, and the path coefficient is a random value in $(-1, 1)$.

$$D_i = |F_j - M_i| \quad (8)$$

Seen from Equation (8), the spiral flight mechanism in the MFO shows robust local and global search capabilities. Thereby, MSISSA balances the algorithm's local and global search abilities by integrating the spiral flight mechanism. The position

update for the followers is calculated using Formula (9).

$$X_{i,j}^{t+1} = \begin{cases} Q \exp\left(\frac{X_{worst}^t - X_{i,j}^t}{i^2}\right) & \text{if } i > \frac{n}{2} \\ D \exp(\cos(2\pi \times c)) + X_{i,j}^t & \text{otherwise} \end{cases} \quad (9)$$

where the coefficient c is a random number in $(-1,1)$, and D represents the distance between the current position $X_{i,j}^t$ and the current discoverers optimal position X_p , as defined in Formula (10).

$$D = |X_{i,j}^t - X_p^t| \quad (10)$$

3.3 Position update combined with Levy flight mechanism for the sentinels

When $f_i > f_g$, the sparrows located at the edge of the population are vulnerable to attacks from predators, and they must converge towards the sparrow at the best position, which diminishes the algorithm's global search capability and increases the likelihood of getting trapped in a local optimum. To overcome this issue, MSISSA combines the Levy flight mechanism with the position update for the sentinels.

Levy flight is a type of random walk where step lengths follow a heavy-tailed probability distribution. Its alternating short and long steps are essential for balancing the local and global search capabilities of optimization algorithms. Short steps enable local search, while occasional long leaps help the algorithm escape local optima. The expression for the Levy flight path formula is defined by (11).

$$Levy(l) = \frac{m}{|n|^{\frac{1}{h}}} \quad (11)$$

where m and n are the parameters that obey the normal distribution: $m: N(0, s_m^2)$, $n: N(0, s_n^2)$. s_m and s_n is calculated by (12).

$$s_m = \frac{\Gamma(1+h) \sin(\rho \pi / 2)}{\Gamma((1+h)/2) 2^{(h-1)/2}} \quad s_n = 1 \quad (12)$$

where $\Gamma(x)$ is the Gamma function. h is a random number in $(0, 2)$, and the empirical value $h = 1.5$.

The random walk mechanism of Levy flight [32-33], due to the uncertainty of its step length, can enhance the algorithm's ability to escape from local optima. Thereby, in MSISSA, the position update for the sentinels using Formula (13).

$$X_{i,j}^{t+1} = \begin{cases} X_{best}^t + b \times |X_{i,j}^t - X_{best}^t| \times Levy(l) & \text{if } f_i > f_g \\ X_{i,j}^t + K \times \frac{|X_{i,j}^t - X_{worst}^t|}{(f_i - f_w) + e} & \text{if } f_i = f_g \end{cases} \quad (13)$$

3.4 Pseudocode Description

The flowchart of MSISSA is shown in Figure 1, and the pseudocode is provided in Algorithm 1.

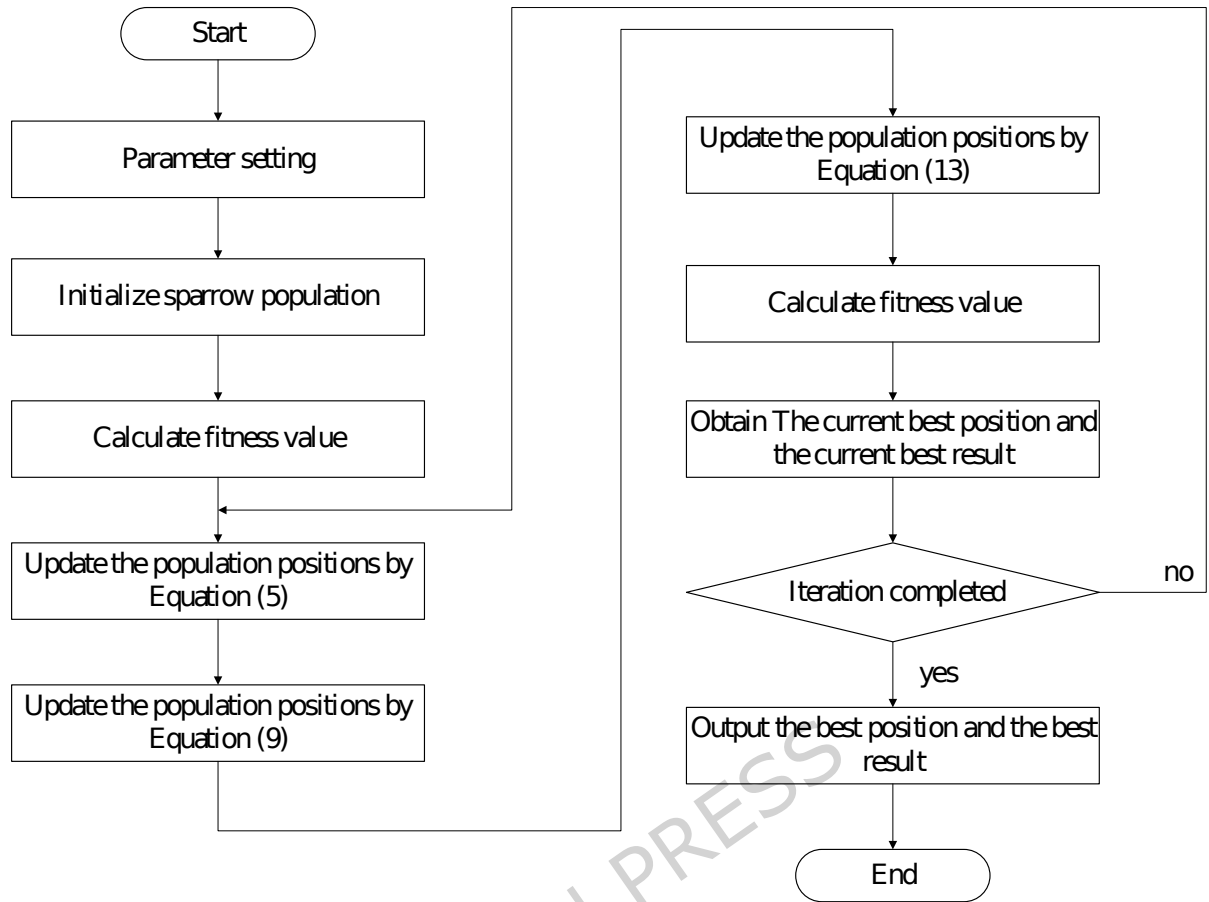


Figure 1. Flowchart of MSISSA

Algorithm 1: MSISSA

Input:

$iter_{max}$: the maximum iterations

PD : the number of the discoverers

SD : the number of the sparrows who perceive the sentinels

R_1 : the alarm value

N : the size of the population

$f(x)$: the object function

Output: X_{best} and f_g

1: Initialize sparrow population $x = [x_1, x_2, \dots, x_N]$

2: Initialize the fitness value list $f_x = [f_{x_1}, f_{x_2}, \dots, f_{x_N}]$

3: **for** $i=1$ to N **do**

4: $f_{x_i} \leftarrow f(x_i)$

5: **end for**

6: Initialize iteration number $t=1$

7: **while** $t < iter_{max}$ **do**

8: Sort $f(x_i)$

9: Sort x_i in ascending order of fitness value

10: $R_1 \leftarrow rand(1)$

11: **for** $i=1$ to PD **do**

12: Update the discoverer position x_i using formula (4)

13: $f_{x_i} \leftarrow f(x_i)$

```

14: end for
15: for  $i = PD + 1$  to  $N$  do
16:   Update the follower position  $X_i$  using formula (9)
17:    $f_{xi} \leftarrow f(X_i)$ 
18: end for
19: for  $i = 1$  to  $SD$  do
20:   Update the sentinel position  $X_i$  using formula (13)
21:    $f_{xi} \leftarrow f(X_i)$ 
22: end for
23:  $t \leftarrow t + 1$ 
24: end while
25: Sort  $f(X_i)$ 
26: Sort  $X_i$  in ascending order of fitness value
27: Return  $X_{best} = X_1, f_g = f(X_1)$ 

```

3.5 Algorithm complexity analysis

Assuming the function has a dimension of d and the population size is N , the worst-case time complexity analysis of the algorithm is as follows.

- (1) The time complexity of population position initialization is $O(d \times N)$.
- (2) The time complexity of calculating the objective function is $O(f(d))$.
- (3) In each iteration, the time complexity of updating the position of the discoverers with Formula (4) is $O(d \times N)$, and the time complexity of calculating the objective function is $O(f(d))$.
- (4) In each iteration, the time complexity of updating the position of the followers with Formula (9) is $O(d \times N)$, and the time complexity of calculating the objective function is $O(f(d))$.
- (5) In each iteration, the time complexity of updating the position of the sentinels with Formula (13) is $O(d \times N \times O(L_{\text{avg}}))$, and the time complexity of calculating the objective function is $O(f(d))$. Where, $O(L_{\text{avg}})$ is taken as $O(1)$.

In summary, the total worst-case time complexity of T iterations are $O(T \times N \times (d + f(d)))$.

4 Experiments and Analyses

4.1 Experimental environment

The experiments were conducted on desktop computer with Gen Intel(R) Core (TM) i5-12400F @2.50 GHz, 16 GB RAM and Window10 operating system, using Python 3.8 as the programming language.

The benchmark functions are usually used to verify performance of algorithm. According to the No Free Lunch Theorem [34], no single algorithm performs optimally across all tasks, necessitating specific analysis for each problem. An algorithm may excel in certain constrained problems but underperform in others. Consequently, a judicious selection of benchmark functions is essential for performance assessment. To evaluate the performance of MSISSA, ten benchmark functions selected from the IEEE CEC Benchmark Suite are shown in Table 1. F_1 - F_3 are dimension-variable unimodal functions with a single extremum, used to evaluate the algorithm's

convergence speed, optimization accuracy, and local search capability. F_4 - F_7 are multimodal functions with multiple local extrema, designed to test the algorithm's ability to escape local optima and the global search capability. F_8 - F_{10} are fixed 2-dimension multimodal functions, chosen to further assess the algorithm's convergence speed, stability, and accuracy.

Table 1. The benchmark functions

No	Name	Function	Dimension	Interval	$f_{\min}$
F_1	Sphere	$f(x) = \sum_{i=1}^d x_i^2$	30,50	(-100,100)	0
F_2	Schwefel 2.22	$f(x) = \sum_{i=1}^d x_i + \prod_{i=1}^d x_i$	30,50	(-10,10)	0
F_3	Schwefel 1.2	$f(x) = \sum_{i=1}^d \sum_{j=1}^d x_j^2$	30,50	(-100,100)	0
F_4	Rastrigin	$f(x) = \sum_{i=1}^d x_i^2 - 10 \cos(2\pi x_i) + 10d$	30,50	(-5.12,5.12)	0
F_5	Ackley	$f(x) = -20 \exp\left(\frac{1}{d} \sum_{i=1}^d \cos(2\pi x_i)\right) + 20 + e$	30,50	(-32, 32)	0
F_6	Griewank	$f(x) = \frac{1}{400} \sum_{i=1}^d x_i^2 - \prod_{i=1}^d \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	30,50	(-600,600)	0
F_7	Penalized	$f(x) = \sin^2(3\pi x_1) + \sum_{i=1}^d (x_i - 1)^2 \sin^2(3\pi x_1) + (x_d - 1)^2 \sin^2(2\pi x_d)$	30,50	(-50,50)	0
F_8	Three Hump Camel	$f(x) = 2x_1^2 - 1.05x_1^4 + \frac{x_1^6}{6} + x_1x_2 + x_2^2$	2	(-5,5)	0
F_9	Schaffer N.2	$f(x) = 0.5 + \frac{\sin^2(x_1 - x_2 - 0.5)}{1 + 0.001(x_1^2 + x_2^2)}$	2	(-100,100)	0
F_{10}	Drop Wave	$f(x) = -\frac{1 + \cos(12\sqrt{x_1^2 + x_2^2})}{0.5(x_1^2 + x_2^2) + 2}$	2	(-5.12,5.12)	-1

4.2 Ablation experiments and analysis

To affirm the viability of MSISSA, ablation studies on the three strategies were performed. The dimensions of functions F_1 - F_7 are set to 30 and 50, while F_8 - F_{10} remain fixed at 2 dimensions. Each of the eight algorithms was run 30 times independently, with a maximum of 500 iterations per run. The eight algorithms involved in ablation experiments are shown in Table

Table 2. The parameter settings for the ablation experiments are provided in

Table

Table 3.

Table 2. Algorithms for ablation experiments

SSA	Standard SSA
SSA1	Algorithm using an adaptive weight operator
SSA2	Algorithm combined with the spiral flight mechanism
SSA3	Algorithm combined with Levy flight mechanism
SSA4	Algorithm combined with an adaptive weight operator and the spiral flight mechanism
SSA5	Algorithm combined with an adaptive weight operator and Levy flight mechanism
SSA6	Algorithm using an adaptive weight operator
MSISSA	Proposed algorithm

Table 3. Parameter setting for eight algorithms

Algorithm	Parameters
SSA	$N = 50, iter_{max} = 500, ST = 0.8, PD = 0.2, SD = 0.1$
SSA1	$N = 50, iter_{max} = 500, ST = 0.8, PD = 0.2, SD = 0.1$
SSA2	$N = 50, iter_{max} = 500, ST = 0.8, PD = 0.2, SD = 0.1$
SSA3	$N = 50, iter_{max} = 500, ST = 0.8, PD = 0.2, SD = 0.1$
SSA4	$N = 50, iter_{max} = 500, ST = 0.8, PD = 0.2, SD = 0.1$
SSA5	$N = 50, iter_{max} = 500, ST = 0.8, PD = 0.2, SD = 0.1$
SSA6	$N = 50, iter_{max} = 500, ST = 0.8, PD = 0.2, SD = 0.1$
MSISSA	$N = 50, iter_{max} = 500, ST = 0.8, PD = 0.2, SD = 0.1$

Table 4. The ablation experimental results (Dim=30)

Function	Index	SSA	SSA1	SSA2	SSA3	SSA4	SSA5	SSA6	MSISSA
F_1	Mean	9.79E-17	1.01E-21	7.58E-18	3.06E-20	1.87E-26	4.87E-24	2.83E-37	5.40E-44
	Std	4.22E-16	3.40E-21	4.08E-17	1.59E-19	1.00E-25	2.46E-23	1.52E-36	2.85E-43
F_2	Mean	5.30E-08	7.67E-10	2.65E-11	7.92E-10	1.61E-13	1.41E-09	3.64E-01	5.14E-20
	Std	2.84E-07	3.94E-09	1.23E-10	2.71E-09	7.39E-13	7.60E-09	1.94E-17	2.76E-19
F_3	Mean	3.07E-15	2.85E-16	1.20E-17	6.20E-21	3.61E-15	1.63E-20	3.40E-40	1.67E-40
	Std	1.13E-14	1.39E-15	6.24E-17	2.81E-20	1.94E-14	8.78E-20	1.50E-39	6.51E-40
F_4	Mean	3.15E-12	2.06E-09	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	Std	1.70E-11	3.37E-13	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F_5	Mean	9.50E-10	1.51E-12	1.84E-12	1.97E-12	1.03E-12	3.88E-12	4.44E-16	4.44E-16
	Std	3.62E-09	6.92E-12	6.92E-12	1.81E-10	4.54E-12	2.01E-11	0.00E+00	0.00E+00
F_6	Mean	5.99E-14	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	Std	3.22E-13	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F_7	Mean	1.44E-08	3.24E-08	2.12E-09	3.33E-08	6.19E-10	6.70E-08	1.04E-08	7.47E-09

Function	Index	SSA	SSA1	SSA2	SSA3	SSA4	SSA5	SSA6	MSISSA
	Std	1.91E-08	4.27E-08	5.27E-09	3.85E-08	1.41E-09	1.16E-07	2.42E-08	3.21E-08

Table 5. The ablation experimental results (Dim=50)

Function	Index	SSA	SSA1	SSA2	SSA3	SSA4	SSA5	SSA6	MSISSA
F_1	Mean	4.13E-17	7.50E-21	2.82E-20	8.66E-21	5.01E-22	8.83E-18	3.97E-41	1.02E-41
	Std	1.57E-16	4.02E-20	1.52E-19	4.06E-20	2.70E-21	4.75E-17	2.14E-40	5.44E-41
F_2	Mean	5.29E-08	3.98E-11	4.80E-10	2.10E-11	1.16E-11	2.25E-10	1.29E-21	1.63E-21
	Std	1.98E-07	1.70E-10	1.80E-09	9.04E-11	6.09E-11	1.21E-09	6.01E-21	5.60E-21
F_3	Mean	4.44E-16	3.22E-17	5.95E-17	1.39E-16	1.25E-22	4.90E-19	9.72E-35	1.87E-42
	Std	1.34E-15	1.73E-16	2.58E-16	5.56E-16	6.66E-22	1.84E-18	5.16E-34	1.00E-41
F_4	Mean	5.68E-13	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	Std	2.99E-12	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F_5	Mean	3.97E-08	2.51E-10	1.91E-10	2.77E-12	3.60E-11	4.63E-14	4.44E-16	4.44E-16
	Std	2.04E-07	1.14E-09	7.35E-10	1.10E-11	1.93E-10	1.42E-13	0.00E+00	0.00E+00
F_6	Mean	1.75E-13	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	Std	9.39E-13	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F_7	Mean	1.18E-07	1.58E-07	7.39E-09	1.72E-07	1.32E-09	5.10E-07	2.51E-08	1.18E-08
	Std	1.91E-07	5.13E-07	2.53E-08	3.15E-07	3.60E-09	8.35E-07	7.15E-08	3.31E-08

Table 6. The ablation experimental results (Dim=2)

Function	Index	SSA	SSA1	SSA2	SSA3	SSA4	SSA5	SSA6	MSISSA
F_8	Mean	7.76E-14	7.96E-24	1.11E-19	3.66E-22	1.87E-26	1.05E-26	8.64E-44	1.87E-46
	Std	4.17E-13	4.29E-23	5.95E-19	1.97E-21	6.84E-26	5.63E-26	3.82E-43	8.84E-46
F_9	Mean	2.96E-16	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	Std	1.39E-15	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F_{10}	Mean	-1.00E+00	-1.00E+00	-1.00E+00	-1.00E+00	-1.00E+00	-1.00E+00	-1.00E+00	-1.00E+00
	Std	1.44E-11	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00

From Tables

[Table 4](#), [Table 5](#) and

[Table 6](#), key observations are:

- (1) On F_1 and F_2 , MSISSA achieves the optimal performance in both convergence accuracy and stability.
- (2) On F_3 in 30 dimensions, SSA6 performs optimal in both convergence accuracy and stability, while on F_3 in 50 dimensions, MSISSA outperforms the others.
- (3) On F_4 - F_6 , MSISSA and SSA6 consistently outperform the other six algorithms in both convergence accuracy and stability.
- (4) On F_7 , SSA4 achieves the optimal convergence accuracy and stability.
- (5) On F_8 , MSISSA demonstrates superior convergence accuracy and stability.

(6) On F_9 and F_{10} , all algorithms (SSA1-SSA6 and MSISSA) show identical convergence accuracy and stability.

Convergence curves on F_1 - F_7 in 30 or 50 dimensions are shown as Figures [Figure 2](#), [Figure 3](#), [Figure 4](#), [Figure 5](#), [Figure 6](#), [Figure 7](#) and [Figure 8](#). Convergence curves on F_8 - F_{10} in 2 dimensions are shown as Figures [Figure 9](#), [Figure 10](#) and [Figure 11](#).

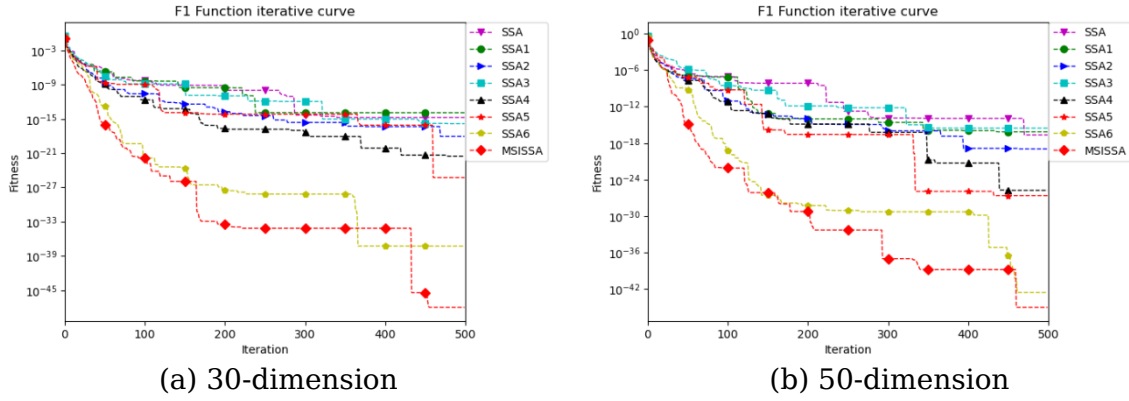


Figure 2. Convergence curves on F_1

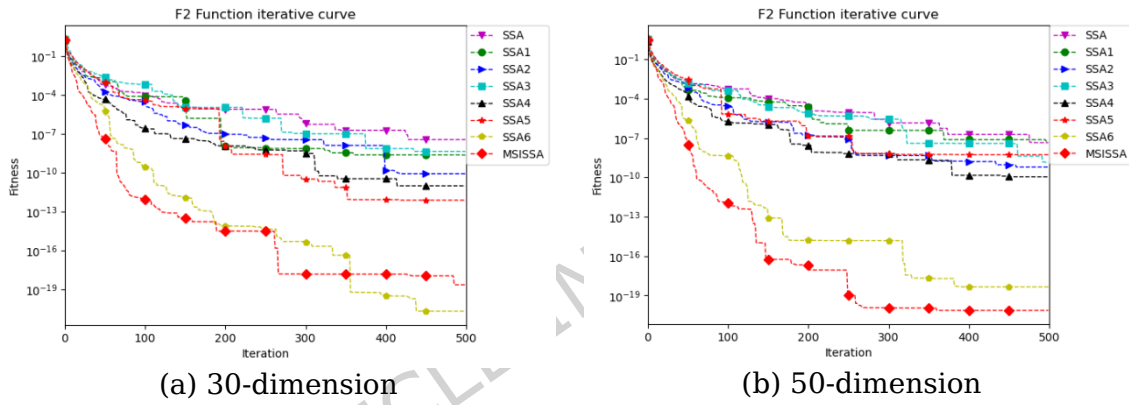


Figure 3. Convergence curves on F_2

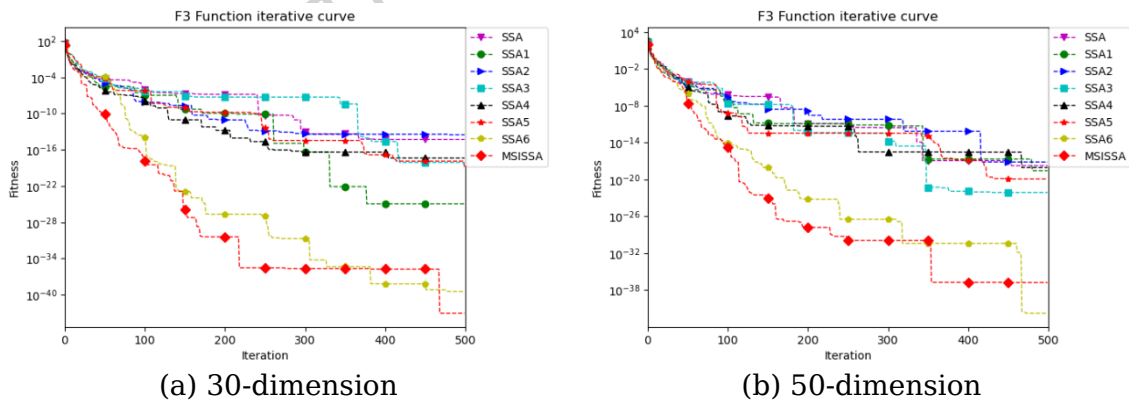
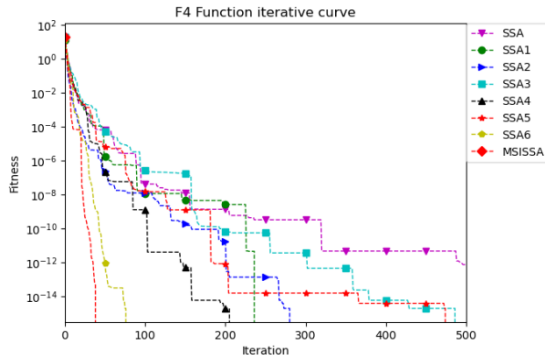
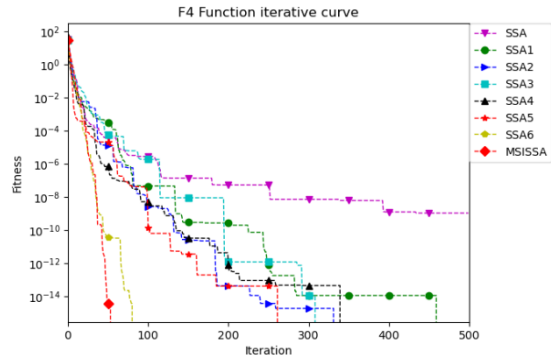


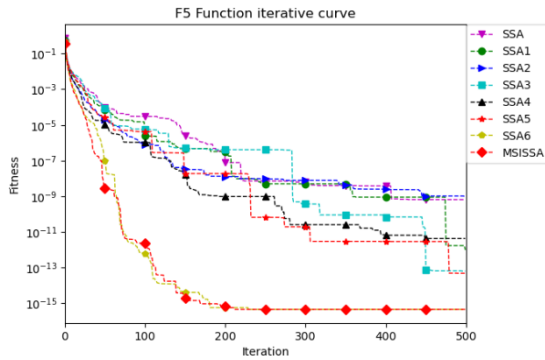
Figure 4. Convergence curves on F_3



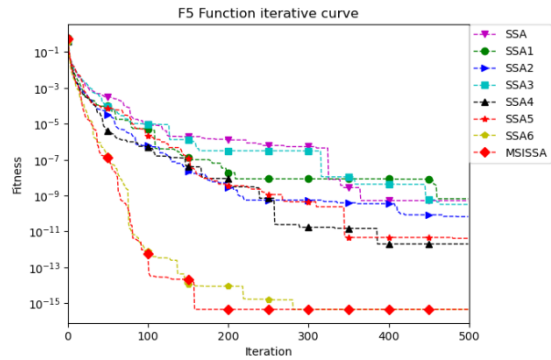
(a) 30-dimension



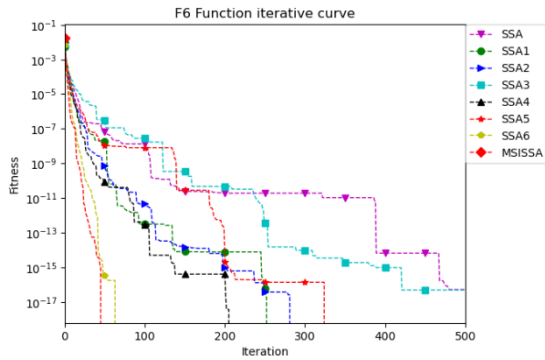
(b) 50-dimension

Figure 5. Convergence curves on F_4 

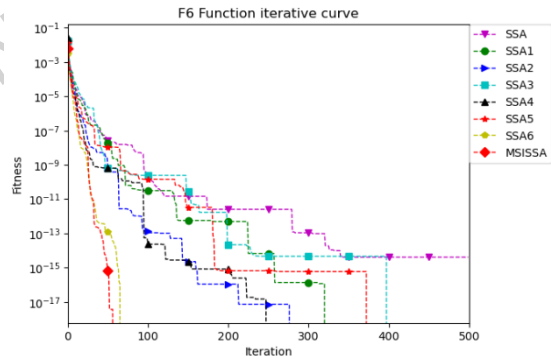
(a) 30-dimension



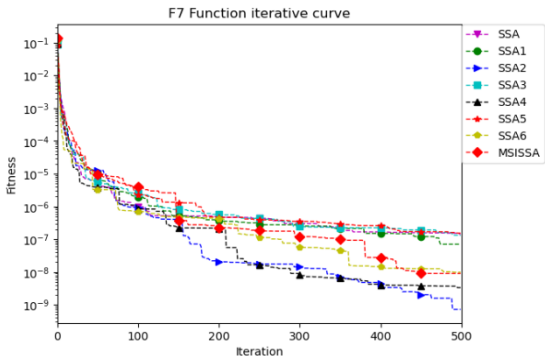
(b) 50-dimension

Figure 6. Convergence curves on F_5 

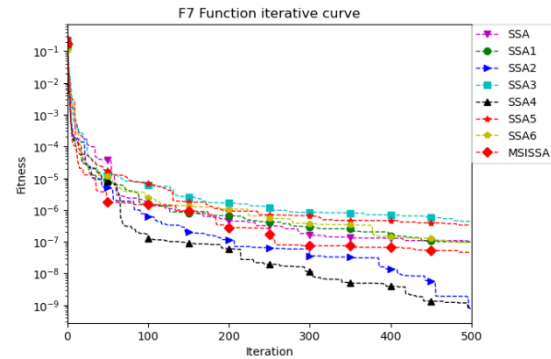
(a) 30-dimension



(b) 50-dimension

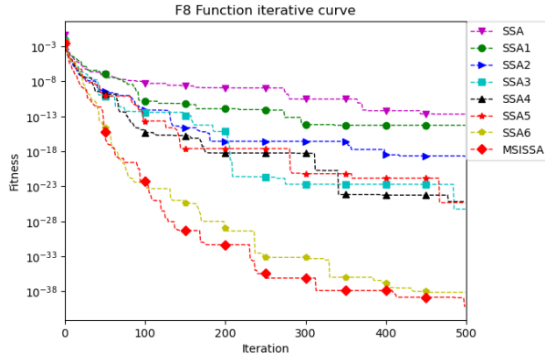
Figure 7. Convergence curves on F_6 

(a) 30-dimension

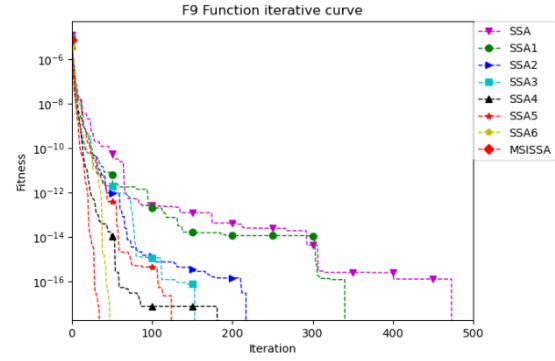


(b) 50-dimension

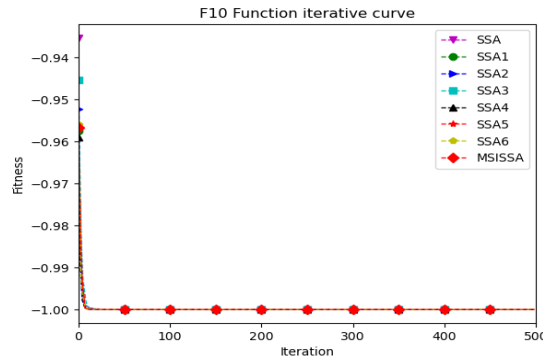
Figure 8. Convergence curves on F_7



(a) 2-dimension

Figure 9. Convergence curves on F8

(a) 2-dimension

Figure 10. Convergence curves on F9

(a) 2-dimension

Figure 11. Convergence curves on F_{10}

From Figures [Figure 2](#), [Figure 3](#), [Figure 4](#), [Figure 5](#), [Figure 6](#), [Figure 7](#), [Figure 8](#), [Figure 9](#), [Figure 10](#) and [Figure 11](#). MSISSA demonstrates the fastest convergence on the unimodal functions F_1 – F_3 and the multimodal functions F_4 – F_6 . On F_7 , MSISSA is slower than SSA2 and SSA4, but faster than the other algorithms. On the 2 dimensions functions F_8 – F_9 , MSISSA outperforms the others. On the 2 dimensions function F_{10} , MSISSA performs comparably to the other algorithms in convergence accuracy.

The ablation experimental results show that the main contributors to the improvement of convergence accuracy are the integration of the spiral flight mechanism and Levy flight mechanism, while the adaptive weight operator accelerates the convergence speed.

4.3 Performance Comparison Experiment and Result Analysis

To verify the performance of the algorithm, MSISSA was compared with the standard SSA[24], four high-performing SSA variants (T_SSA [25], LSSA [26], T_TentSSA [27], CASSA [28]), and two classical algorithms (PSO [1], MFO [4]) on functions F_1 – F_{10} . Each algorithm was run 30 times, with a maximum of 500 iterations per run. The parameter settings for algorithms are listed in [Table 7](#).

Table 7. Parameter settings for algorithms

algorithms	Parameter
MFO [4]	$N = 50, iter_{max} = 500, r = 2$
PSO [1]	$N = 50, iter_{max} = 500, c_1 = 0.5, c_2 = 0.5$
SSA [24]	$N = 50, iter_{max} = 500, ST = 0.8, PD = 0.2, SD = 0.1$
CASSA [25]	$N = 50, iter_{max} = 500, ST = 0.8, PD = 0.2, SD = 0.1$
T_TentSSA [26]	$N = 50, iter_{max} = 500, ST = 0.8, PD = 0.2, SD = 0.1$

LSSA [27]	$N = 50, iter_{max} = 500, ST = 0.8, PD = 0.2, SD = 0.1$
T_SSA [28]	$N = 50, iter_{max} = 500, ST = 0.8, PD = 0.2, SD = 0.1$
MSISSA	$N = 50, iter_{max} = 500, ST = 0.8, PD = 0.2, SD = 0.1$

The performance comparison experimental results for eight algorithms on ten benchmark functions are detailed in Tables [Table 8](#), [Table 9](#) and [Table 10](#).

Table 8. The performance comparison experimental results (Dim=30)

Function	Index	MFO	PSO	SSA	CASSA	T_TentS SA	LSSA	T_SSA	MSISSA
F_1	Mean	6.04E+04	6.45E+04	2.19E-14	9.92E-22	1.57E-18	4.69E-16	1.42E-14	1.97E-40
	Std	8.64E+03	5.94E+03	1.09E-13	3.72E-21	7.89E-18	1.79E-15	7.62E-14	1.06E-39
F_2	Mean	2.11E+02	1.87E+12	1.20E-08	7.24E-12	2.96E-09	4.59E-07	5.69E-08	1.18E-21
	Std	1.39E+01	4.35E+12	5.90E-08	2.85E-11	8.21E-09	2.22E-06	3.02E-07	6.20E-21
F_3	Mean	9.86E+04	1.31E+05	4.53E-10	9.97E-22	1.22E-14	7.43E-14	3.04E-12	3.28E-43
	Std	1.25E+04	3.16E+04	2.41E-09	3.67E-21	6.53E-14	2.51E-13	1.63E-11	1.25E-42
F_4	Mean	4.21E+02	4.30E+02	3.79E-15	1.89E-15	0.00E+00	1.89E-15	1.33E-14	0.00E+00
	Std	2.58E+01	3.14E+01	2.04E-14	1.02E-14	0.00E+00	1.02E-14	3.18E-14	0.00E+00
F_5	Mean	2.00E+01	2.00E+01	6.82E-08	6.50E-11	4.49E-08	6.98E-08	1.13E-08	4.44E-16
	Std	5.10E-03	1.09E-03	2.66E-07	3.48E-10	1.78E-07	3.68E-07	4.30E-08	0.00E+00
F_6	Mean	5.92E+02	5.80E+02	2.44E-15	0.00E+00	0.00E+00	1.13E-15	0.00E+00	0.00E+00
	Std	5.46E+01	6.03E+01	1.31E-14	0.00E+00	0.00E+00	4.39E-15	0.00E+00	0.00E+00
F_7	Mean	9.68E+08	1.03E+09	4.51E-08	1.29E-10	1.87E-07	8.34E-08	1.26E-07	1.69E-09
	Std	1.78E+08	1.64E+08	9.64E-08	2.16E-10	3.13E-07	1.55E-07	1.96E-07	4.38E-09

Table 9. The performance comparison experimental results (Dim=50)

Function	Index	MFO	PSO	SSA	CASSA	T_TentSSA	LSSA	T_SSA	MSISSA
F_1	Mean	1.21E+05	1.20E+05	3.56E-17	7.67E-22	3.20E-12	1.43E-14	3.36E-23	5.18E-40
	Std	8.54E+03	6.87E+03	1.91E-16	4.13E-21	1.72E-11	7.66E-14	1.71E-22	2.79E-39
F_2	Mean	1.47E+03	2.93E+22	5.39E-08	9.90E-12	2.52E-07	3.75E-08	1.85E-10	8.63E-22
	Std	4.58E+03	6.85E+22	2.87E-07	4.42E-11	1.35E-06	9.98E-08	8.95E-10	4.40E-21
F_3	Mean	2.53E+05	3.48E+05	2.14E-14	5.36E-16	3.23E-14	9.56E-14	1.02E-13	2.91E-44
	Std	3.51E+04	1.18E+05	1.15E-13	1.68E-15	1.66E-13	4.07E-13	5.31E-13	1.51E-43
F_4	Mean	7.49E+02	7.58E+02	1.81E-11	2.27E-14	1.06E-13	0.00E+00	1.14E-14	0.00E+00
	Std	3.52E+01	2.72E+01	9.76E-11	1.22E-13	4.69E-13	0.00E+00	4.50E-14	0.00E+00
F_5	Mean	2.00E+01	2.00E+01	1.90E-09	1.95E-11	2.62E-09	1.60E-09	3.71E-08	4.44E-16
	Std	2.86E-02	6.89E-04	7.16E-09	7.36E-11	1.09E-08	7.00E-09	1.95E-07	0.00E+00
F_6	Mean	1.07E+03	1.10E+03	0.00E+00	0.00E+00	0.00E+00	3.63E-16	0.00E+00	0.00E+00
	Std	6.85E+01	7.16E+01	0.00E+00	0.00E+00	0.00E+00	1.70E-15	0.00E+00	0.00E+00
F_7	Mean	2.15E+09	2.02E+09	1.31E-07	2.84E-10	8.00E-08	1.20E-07	2.02E-07	4.22E-08
	Std	2.48E+08	3.20E+08	2.31E-07	4.43E-10	1.16E-07	2.32E-07	3.69E-07	1.27E-07

Table 10. The performance comparison experimental results (Dim=2)

Function	Index	MFO	PSO	SSA	CASSA	T_TentSSA	LSSA	T_SSA	MSISSA
F_8	Mean	6.60E-03	5.13E-01	4.99E-12	1.48E-20	6.30E-13	1.21E-15	4.30E-14	7.49E-44
	Std	2.88E-02	3.00E-01	2.05E-11	7.82E-20	3.29E-12	5.22E-15	2.32E-13	4.03E-43
F_9	Mean	4.34E-02	2.68E-01	1.85E-16	0.00E+00	0.00E+00	1.48E-17	0.00E+00	0.00E+00
	Std	5.15E-02	1.13E-01	7.39E-16	0.00E+00	0.00E+00	5.54E-17	0.00E+00	0.00E+00
F_{10}	Mean	9.39E-01	6.97E-01	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00
	Std	2.50E-02	1.47E-01	8.61E-10	0.00E+00	4.75E-10	5.89E-15	7.48E-13	0.00E+00

From Tables [Table 8](#), [Table 9](#) and

[Table 10](#), the following conclusions can be drawn:

- (1) On F_1 - F_3 , both in 30 and 50 dimensions, MSISSA demonstrates superior convergence accuracy and better stability compared to the other seven algorithms.
- (2) On F_4 , both in 30 and 50 dimensions, MSISSA outperforms SSA, T_SSA, T_TentSSA, CASSA, PSO and MFO, achieves the same performance as LSSA, and can converge to the theoretical optimal value of 0.
- (3) On F_5 , both in 30 and 50 dimensions, MSISSA outperforms the other seven algorithms in both convergence accuracy and stability.
- (4) On F_6 , both in 30 and 50 dimensions, MSISSA surpasses MFO, PSO, and LSSA in both accuracy and stability, while matching the performance of the other four algorithms, and can converge to the theoretical optimal value of 0.

- (5) On F_7 , both in 30 and 50 dimensions, MSISSA performs worse than CASSA, but better than the others.
- (6) On F_8 - F_9 , MSISSA excels in both convergence accuracy and stability compared to all other algorithms, and converges to the theoretical optimal value of 0.
- (7) On F_{10} , MSISSA matches the performance of SSA, CASSA, T_TentSSA, and LSSA in both accuracy and stability, outperforming MFO and PSO, and can converge to the theoretical optimal value of -1.

Convergence curves on F_1 - F_7 in 30 or 50 dimensions are shown as Figures [Figure 12](#), [Figure 13](#), [Figure 14](#), [Figure 15](#), [Figure 16](#), [Figure 17](#) and [Figure 18](#). Convergence curves on F_8 - F_{10} in 2 dimensions are shown as Figures [Figure 19](#), [Figure 20](#) and [Figure 21](#).

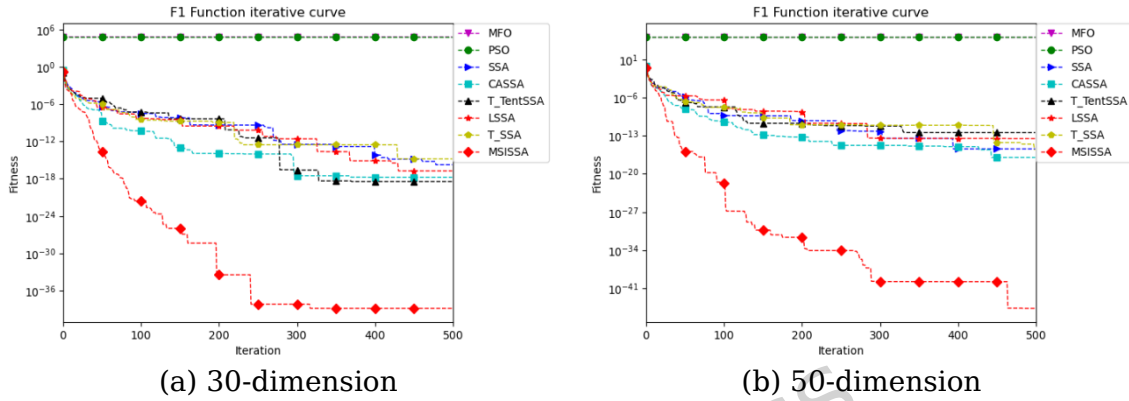


Figure 12. Convergence curves on F_1

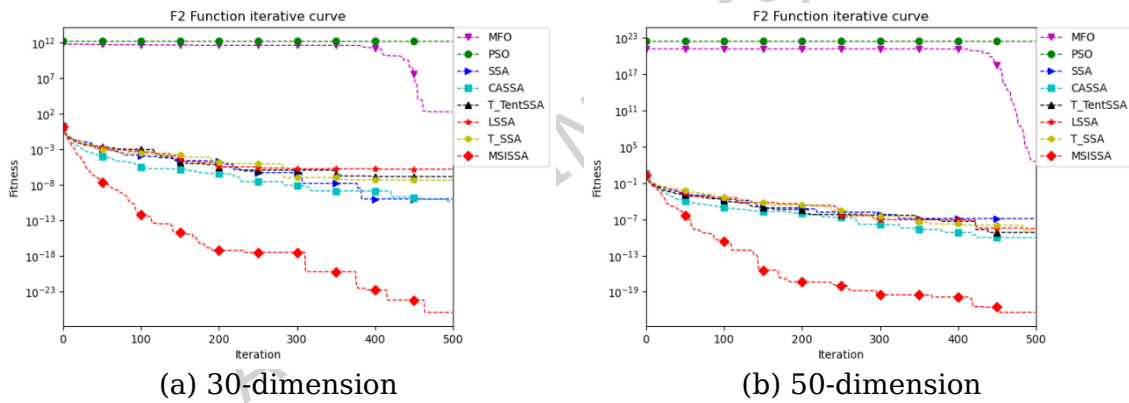


Figure 13. Convergence curves on F_2

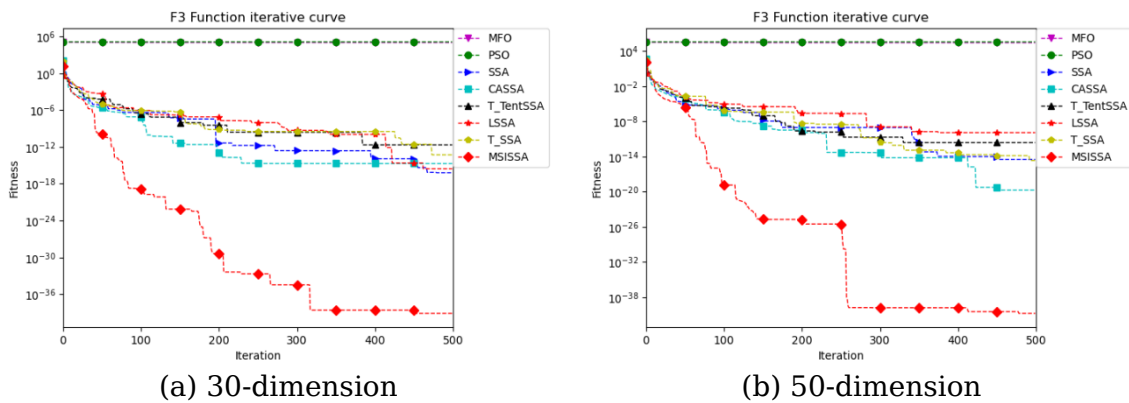
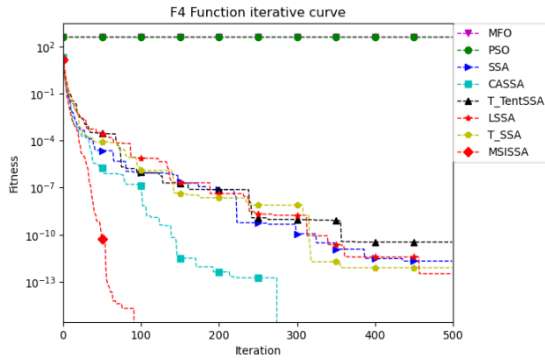
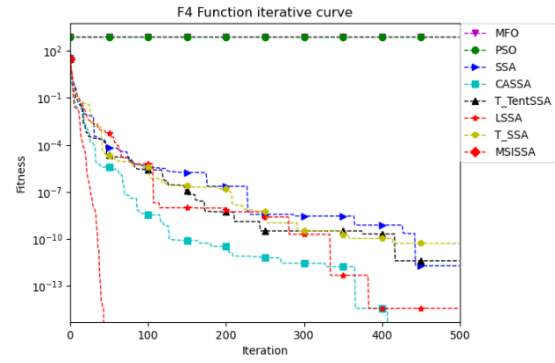


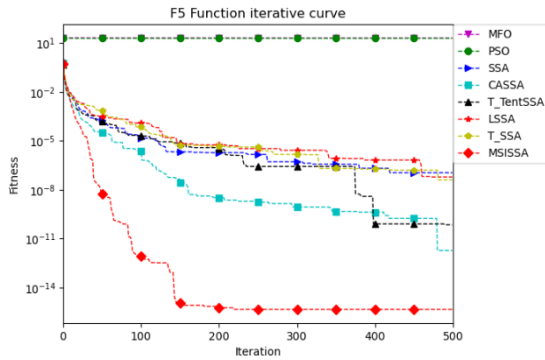
Figure 14. Convergence curves on F_3



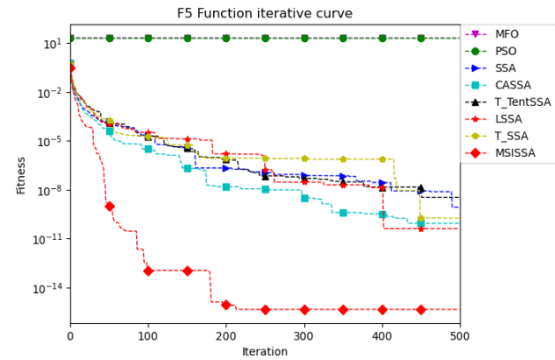
(a) 30-dimension



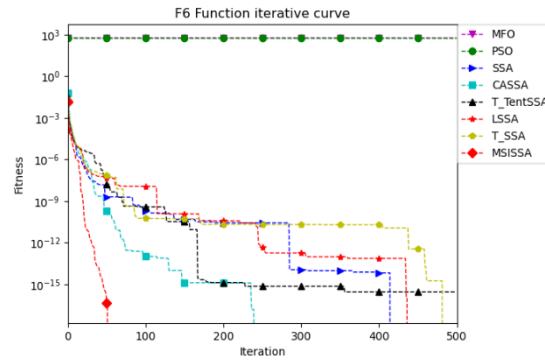
(b) 50-dimension

Figure 15. Convergence curves on F_4 

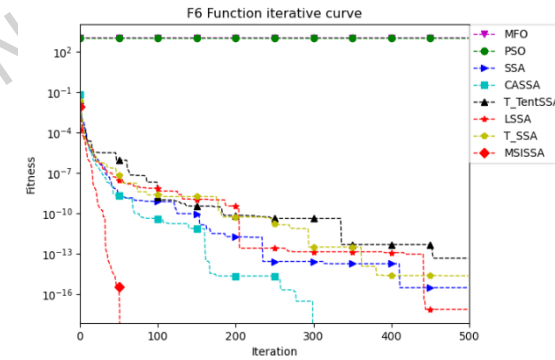
(a) 30-dimension



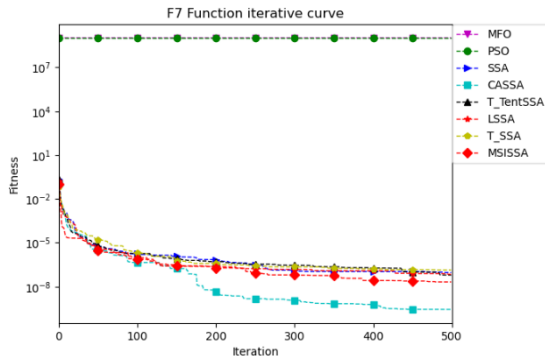
(b) 50-dimension

Figure 16. Convergence curves on F_5 

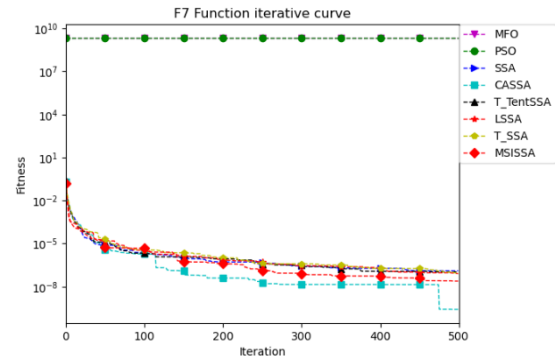
(a) 30-dimension



(b) 50-dimension

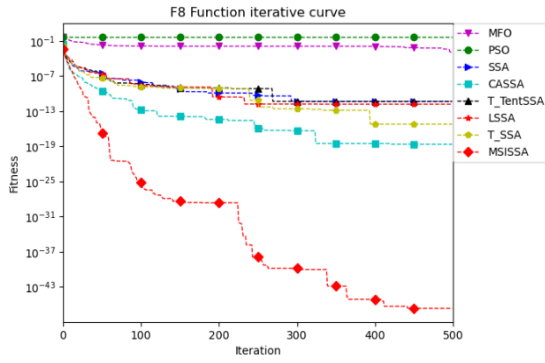
Figure 17. Convergence curves on F_6 

(a) 30-dimension

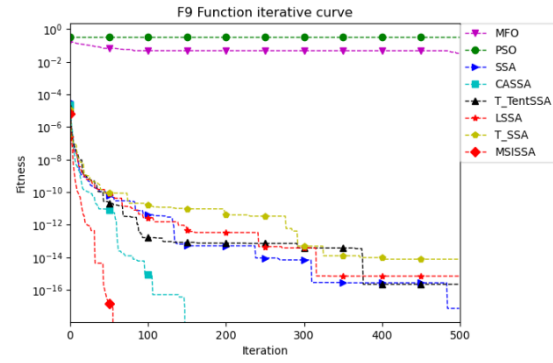


(b) 50-dimension

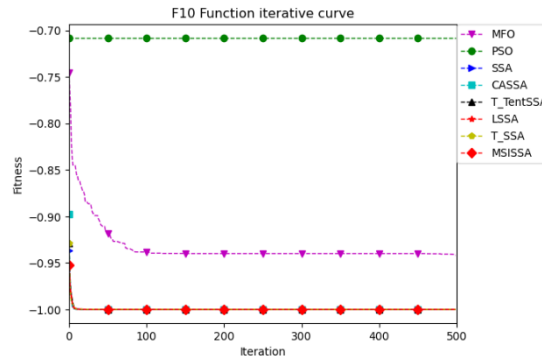
Figure 18. Convergence curves on F_7



2-dimension

Figure 19.Convergence curves on F_8 

2-dimension

Figure 20.Convergence curves on F_9 

2-dimension

Figure 21.Convergence curves on F_{10}

From Figures [Figure 12](#), [Figure 13](#), [Figure 14](#), [Figure 15](#), [Figure 16](#), [Figure 17](#), [Figure 18](#), [Figure 19](#), [Figure 20](#), and [Figure 21](#). on F_1 - F_6 and F_8 - F_9 , MSISSA demonstrates faster convergence than the others. On F_7 , MSISSA is superior to the other compared algorithms except CASSA. On F_{10} , MSISSA shows comparable convergence speed to SSA, CASSA, T_TentSSA, LSSA, and T_SSA, while outperforming MFO and PSO. MFO and PSO exhibit higher initial fitness values because they generate initial solutions through completely random initialization, causing these random solutions to fall far from the optimal solution region, which results in elevated initial objective function values.

4.4 Wilcoxon Test

Based on the experimental data in Sections 4.2 and 4.3, the Wilcoxon rank-sum test [35] is employed to further evaluate whether there are significant differences among MSISSA, MFO, PSO, SSA, CASSA, T_TentSSA, LSSA, and T_SSA. The results of the rank-sum test on F_1 - F_{10} are presented in Table 11. In the table, the "W" column shows the comparison outcomes, where the symbols '+', '-', and '=' indicate that MSISSA outperforms, underperforms, or performs equally to the compared algorithm, respectively.

Table 11. The results of the Wilcoxon rank sum test

Function	Dim	MFO	PSO	SSA	CASSA	T_TentSSA	LSSA	T_SSA
F_1	30	+	+	+	+	+	+	+
	50	+	+	+	+	+	+	+
F_2	30	+	+	+	+	+	+	+
	50	+	+	+	+	+	+	+
F_3	30	+	+	+	+	+	+	+

Function	Dim	MFO	PSO	SSA	CASSA	T_TentSSA	LSSA	T_SSA
F_4	50	+	+	+	+	+	+	+
	30	+	+	+	+	=	+	+
	50	+	+	+	+	+	=	+
F_5	30	+	+	+	+	+	+	+
	50	+	+	+	+	+	+	+
	30	+	+	+	=	=	+	=
F_6	50	+	+	=	=	=	+	=
	30	+	+	+	-	+	+	+
	50	+	+	+	-	+	+	+
F_8	2	+	+	+	+	+	+	+
F_9	2	+	+	+	=	=	+	+
F_{10}	2	+	+	=	=	=	=	=
+		17	17	15	11	12	15	14
-		0	0	0	2	0	0	0
=		0	0	2	4	5	2	3

From

Table 11, the number of '+' significantly exceeds the number of '-' and '=', indicating that MSISSA is superior to the other seven algorithms in terms of optimization stability and convergence accuracy.

4.5 Application to Classical Engineering Problems

The MSISSA was applied to three classical engineering problems. Experiments were conducted to compare its results with those of other optimization algorithms, further validating its feasibility and effectiveness. To objectively reflect the experimental results, the parameter settings for each algorithm were kept consistent with the aforementioned comparative tests. Each of the eight algorithms was independently run 30 times, with a maximum of 500 iterations per run. The simulation results for the three classical engineering problems are presented in Table

Table 12.

Case 1: Spring Design

The objective of the tension/compression spring design problem is to minimize the spring weight while satisfying constraints on minimum deflection, vibration frequency, and shear stress. This problem involves three continuous decision variables: wire diameter (x_1), mean coil diameter (x_2), and number of active coils (x_3). The objective function for this problem is defined as,

$$\min f(x) = (x_3 + 2) x_1 x_2 \quad (14)$$

Constraints

$$y_1 = 1 - \frac{x_2^3 x_3}{71785 x_1^4} \leq 0 \quad (15)$$

$$y_2 = \frac{4x_2^2 - x_1 x_2}{12566(x_1^3 x_2 - x_1^4)} + \frac{1}{5108 x_1^2} - 1 \leq 0 \quad (16)$$

$$y_3 = 1 - \frac{140.45 x_1}{x_2^2 x_3} \leq 0 \quad (17)$$

$$y_4 = \frac{x_1 + x_2}{1.5} - 1 \leq 0 \quad (18)$$

where $0.05 \leq x_1 \leq 2$, $0.25 \leq x_2 \leq 1.3$, $2 \leq x_3 \leq 15$.

Case 2: Three-Bar Truss Design

In the three-bar truss design problem, variables x_1 , x_2 and x_3 represent the cross-sectional areas of the three bars, respectively. Due to symmetry, $x_1 = x_3$. Thus, the objective of this design can be described as: minimizing the volume of the truss by adjusting the cross-sectional areas (x_1, x_2). The truss is subject to stress constraints (s)

on each structural member. The objective function for this problem is defined as,

$$\min f(x) = (2\sqrt{2}x_1 + x_2)/l \quad (19)$$

Constraints

$$y_1 = \frac{\sqrt{2}x_1 + x_2}{\sqrt{2}x_1^2 + 2x_1x_2} P - s \leq 0 \quad (20)$$

$$y_2 = \frac{x_2}{\sqrt{2}x_1^2 + 2x_1x_2} P - s \leq 0 \quad (21)$$

$$y_3 = \frac{1}{\sqrt{2}x_2 + x_1} P - s \leq 0 \quad (22)$$

where $0.001 \leq x_1 \leq 1$, $0.001 \leq x_2 \leq 1$, $l = 100\text{cm}$, $P = 2\text{KN}/\text{cm}^2$, $s = 2\text{KN}/\text{cm}^2$.

Case 3: Pressure Vessel Design

The objective of the pressure vessel design is to minimize the total manufacturing cost (including material, forming, and welding costs). The cylindrical pressure vessel is sealed at both ends by heads, with one head being hemispherical. The four optimization variables for this problem are: L length of the cylindrical section (excluding heads), R Inner radius of the cylindrical shell, T_s Thickness of the cylindrical wall, T_h Thickness of the hemispherical head. x_1, x_2, x_3, x_4 represent T_s, T_h, R, L respectively. The objective function is defined as,

$$\min f(x) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1x_3^2 \quad (23)$$

Constraints

$$y_1 = -x_1 + 0.0193x_3 \leq 0 \quad (24)$$

$$y_2 = -x_2 + 0.00954x_3 \leq 0 \quad (25)$$

$$y_3 = -\rho x_3^2 - 4\rho x_3^3/3 + 129600 \leq 0 \quad (26)$$

$$y_4 = x_4 - 240 \leq 0 \quad (27)$$

where $0 \leq x_1 \leq 100$, $0 \leq x_2 < 100$, $10 \leq x_3 \leq 100$, $10 \leq x_4 \leq 100$.

Experiments Results Analysis

The results of the algorithm on three classical engineering problems are presented in Table 10.

Table 12. Experiments Results

Cas e	Index	MFO	PSO	SSA	CASSA	T_TentS SA	LSSA	T_SSA	MSISS A
1	Mae n	1.97E+04	5.48E+05	1.10E+04	1.65E+04	9.34E+03	8.73E+03	9.43E+03	8.18E+03
	std	1.64E+04	3.37E+05	1.02E+03	1.20E+03	7.23E+02	3.42E+02	8.17E+02	5.32E+01
	min	8.07E+03	1.15E+05	9.49E+03	1.34E+04	8.34E+03	8.23E+03	8.34E+03	8.07E+03
2	Mae n	2.65E+02	2.73E+02	2.67E+02	2.68E+02	2.64E+02	3.49E+02	2.64E+02	2.64E+02
	std	9.89E-01	7.33E+00	2.51E-02	2.82E+00	5.59E-02	2.40E+01	6.21E-02	4.47E-02
	min	2.64E+02	2.64E+02	2.64E+02	2.64E+02	2.64E+02	3.81E+02	2.64E+02	2.64E+02
3	Mae	1.97E+04	5.48E+05	1.10E+04	1.65E+04	9.34E+03	8.73E+03	9.43E+03	8.18E+03

	n	04	05	04	04		03	03	03
	std	1.64E+04	3.37E+05	1.02E+03	1.20E+03	7.23E+02	3.42E+02	8.17E+02	5.32E+01
	min	8.07E+03	1.15E+05	9.49E+03	1.34E+04	8.34E+03	8.23E+03	8.34E+03	8.07E+03

As shown in

Table 12, the experimental results indicate that MSISSA outperforms all other algorithms across three classical engineering applications. To further validate the efficiency of MSISSA, it is applied to 3D wireless sensor deployment.

5 Application in the Deployment of 3D-WSNs

3D-WSNs have shown wide application potential in various fields such as drone swarm flight [36], environmental monitoring [37], and 3D WSN localization [38-39]. However, as the network scale continues to expand, the issue of node deployment has become increasingly prominent, posing a key technological challenge that restricts network coverage and performance improvement. This manuscript applies MSISSA to optimize the deployment of 3D-WSNs, focusing on improving the coverage of wireless sensor networks.

5.1 Model of the Deployment of 3D-WSNs

When discussing the deployment of 3D-WSNs, it is assumed that each node is connected via a wireless network. The monitoring range of each sensor in 3D space is abstracted as a sphere, with the sensor's position serving as the center and the monitoring distance R_s as the radius.

Assume the deployment area is a 3D space with dimension $L' W' H$, and need deploy N sensor nodes. Any point in the 3D space is denoted as $g = (b, c, e)$, and the position of sensor node is denoted as $S_i = (x_i, y_i, z_i)$. Each node has the same sensing radius R_s . If a target point g is located within the sensing range of the node S_i , it means g can be detected by S_i . The Euclidean distance [40] between g and S_i is calculated as,

$$d(g, S_i) = \sqrt{(b - x_i)^2 + (c - y_i)^2 + (e - z_i)^2} \quad (28)$$

If point g is within the monitoring range of the node S_i , the coverage probability is 1; otherwise, it is 0. The coverage probability is calculated as,

$$p(g, S_i) = \begin{cases} 1 & d(g, S_i) \leq R_s \\ 0 & d(g, S_i) > R_s \end{cases} \quad (29)$$

If there are N sensor nodes in the space, the coverage of point g by the sensors is calculated as,

$$p(g, S) = 1 - \prod_{i=1}^N [1 - p(g, S_i)] \quad i=1, 2, \dots, N \quad (30)$$

The overall coverage of 3D-WSNs represents the average coverage of all target points in the space. If the coverage of the network is denoted as COV , the following Formula (31) can act as the objective function for optimizing the deployment of 3D-WSNs.

$$COV = \frac{\sum_{i=1}^{W' H' L} \sigma(g, S_i)}{W' H' L} \quad (31)$$

5.2 Simulation Experiments

In the same experimental environment, the deployment of 3D-WSNs is optimized using the MSISSA and compared with standard SSA, CASSA, T_TentSSA, LSSA, T_SSA, PSO, MFO. The COV of the 3D-WSNs is to evaluate the superiority and applicability of MSISSA. Each algorithm was run independently 10 times, with a maximum of 100 iterations. The parameter settings for the 3D-WSNs coverage model are as follows: N is 30 and 50, $W=50, L=50, H=50, R_s=13$.

The experimental results are shown in Table 13.

Table 13. The simulation experimental results

Algorithm	Performance Index	Number of Nodes 30	Number of Nodes 50
MFO	Mean	84.83%	94.83%
	Best	87.19%	95.18%
	Std	1.477E-02	3.161E-03
PSO	Mean	85.20%	94.16%
	Best	89.03%	94.87%
	Std	2.266E-02	4.879E-03
SSA	Mean	88.60%	96.85%
	Best	90.28%	97.16%
	Std	1.236E-02	2.940E-03
CASSA	Mean	88.11%	96.55%
	Best	88.82%	97.48%
	Std	6.068E-03	8.194E-03
T_TentSSA	Mean	80.11%	92.78%
	Best	84.57%	94.83%
	Std	4.756E-02	2.081E-02
LSSA	Mean	87.44%	96.47%
	Best	90.04%	97.39%
	Std	1.937E-02	1.029E-02
T_SSA	Mean	86.35%	96.28%
	Best	87.23%	97.15%
	Std	8.564E-03	8.040E-03
MSISSA	Mean	88.94%	96.86%
	Best	91.89%	99.11%
	Std	1.266E-02	1.081E-02

From Table 13, when the number of sensor nodes is 30, MSISSA outperforms MFO, PSO, SSA, CASSA, T_TentSSA, LSSA, and T_SSA in terms of maximum coverage rate, achieving improvements of 4.69%, 2.86%, 1.60%, 3.07%, 7.32%, 1.85%, and 4.66%, respectively. Additionally, MSISSA's average coverage rate is increased by 4.11%, 3.73%, 0.33%, 0.83%, 8.83%, 1.50%, and 2.59%. When the number of sensor nodes is 50, MSISSA also demonstrates superiority over MFO, PSO, SSA, CASSA, T_TentSSA, LSSA, and T_SSA in terms of maximum coverage rate, with improvements of 3.94%, 4.24%, 1.95%, 1.63%, 4.29%, 1.72%, and 1.96%, respectively. Furthermore,

MSISSA's average coverage rate is enhanced by 2.02%, 2.70%, 0.01%, 0.30%, 4.08%, 0.39%, and 0.58%. Obviously, MSISSA exhibits better performance in optimizing the deployment of 3D-WSNs than the other algorithms.

The convergence curves are depicted in Figure 22.

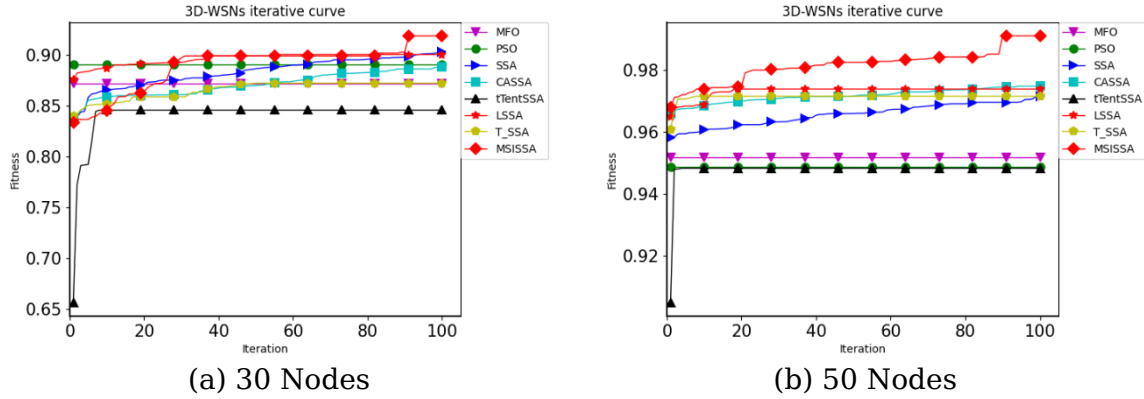


Figure 22. Convergence curves

From Figure 22. Convergence curves, compared to the other seven algorithms, MSISSA exhibits superior convergence accuracy and faster convergence speed, has better ability to jump out of the local optimal and find the global optimal, can more effectively improve 3D-WSNs coverage, make node distribution more uniform, and reduce coverage redundancy.

In summary, MSISSA has obvious advantages to optimize the deployment of 3D-WSNs and reduce deployment costs.

6 Conclusions

To address the shortcomings of the standard SSA in solving high-dimensional problems, such as slow convergence accuracy and convergence speed, poor global search capability, and susceptibility to local optima, this manuscript proposes an improved algorithm named MSISSA, and then applied MSISSA to optimize the deployment of 3D-WSNs. MSISSA integrates three strategies: an adaptive weight operator, the spiral flight mechanism of MFO, and the Levy flight mechanism. First, MSISSA enhances the convergence speed by introducing an adaptive weight operator ω with exponential decay to the position update formula for the discoverers. Second, MSISSA balances the algorithm's local and global search abilities by combining the spiral flight mechanism of the MFO into the position update formula for the followers. Third, MSISSA escapes local optima by using the Levy flight mechanism with position updates for the sentinels. The ablation experiments, performance comparison experiments, and validation on optimization of classical engineering problems have verified the superiority of MSISSA, and the simulation experiments have confirmed its applicability and stability.

Future research aims to further improve the applicability and robustness of MSISSA across diverse environments and under various constraints. Furthermore, plans include utilizing MSISSA to optimize deep learning models in subsequent work.

Author contributions

Shan Li conceived and directed the research, managed data, and wrote the manuscript draft; Liping Mo provided the methodology and participated in manuscript editing; Biao Yin processed the data; Wei Min conducted data validation and visualization. All authors reviewed and approved the manuscript.

Funding:

This work was supported by the National Natural Science Foundation of China (Grant No. 62266019) and the Natural Science Foundation of Hunan Province, China (Grant No. 2024JJ7394).

Data availability:

Data are contained within the manuscript.

Conflict of interest:

The authors declare no competing interests.

Additional information □

Correspondence and requests for materials should be addressed to L.M.

Reprints and permissions information is available at www.nature.com/reprints.

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

References

1. Kennedy J, Eberhart R. Particle swarm optimization. *Icnn95-international Conference on Neural Networks. IEEE1995* (1995).
2. Koza J R. Hierarchical genetic algorithms operating on populations of computer programs. *International Joint Conference on Artificial Intelligence. Morgan Kaufmann Publishers Inc. 1989* (1989).
3. Gandomi A H, Yang X S, Alavi A H. Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems. *Engineering with Computers* 29(2), 245-245 (2013).
4. Mirjalili S. Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm. *Knowledge-based systems* 89, 228-249 (2015).
5. Saremi S, Mirjalili S, Lewis A. Grasshopper optimization algorithm: Theory and application. *Advances in Engineering Software* 105, 30-47 (2017).
6. Mirjalili S. How effective is the grey wolf optimizer in training multi-layer perceptron. *Applied intelligence* 43, 150-161 (2015).
7. Chen Z, Francis A, Li S, et al. Egret swarm optimization algorithm: an evolutionary computation approach for model free optimization. *Biomimetics* 7(4), 144 (2022).
8. Wang Z H, Dai Y Q, Liu H. 3D path planning based on an adaptive moth-flame optimization algorithm. *Application Research of Computers* 40(1), 7 (2023).
9. Min W, Mo L, Yin B, Li S. An improved cuckoo search algorithm and its application in robot path planning. *Applied Sciences* 14(20), 2076-3417 (2024).
10. Ou Y, Yin P, Mo L P. An improved grey wolf optimizer and its application in robot path planning. *Biomimetics* 8(1), 84 (2023).
11. Yin B, Mo L, Min W, et al. An improved beetle antennae search algorithm and its application in coverage of wireless sensor networks. *Scientific Reports* 14(1), 29372 (2024).
12. Shen X H, Ma X J, Niu S H. Optimization of pipeline sensor network coverage based on improved lion herding algorithm. *Journal of Sensing Technology* 35(09), 1268-1276 (2022).
13. Rawat P, Singh K D, Chao C H, Bonnin J M. Wireless sensor networks: a survey on recent developments and potential synergies. *The Journal of supercomputing* 68, 1-48 (2014).
14. Wu W, et al. A label-based ant colony algorithm for heterogeneous vehicle routing with mixed backhaul. *Applied Soft Computing* 47, 224-234 (2016).
15. Liu J, Qu C, Zhang L, et al. A new hybrid algorithm for three-stage gene selection based on whale optimization. *Scientific Reports* 13(1), 3783 (2023).
16. Huang Z, Liao B, Cao X, et al. Adopting nonlinear activated beetle antennae search algorithm for fraud detection of public trading companies: a computational finance approach. *Mathematics* 10(13), 2160 (2022).
17. Ye S Q, Zhou K Q, Zhang C X, et al. An improved multi-objective cuckoo search

- approach by exploring the balance between development and exploration. *Electronics* 11(5), 704 (2022).
18. Ou Y, Qin F, Zhou K Q, et al. An improved grey wolf optimizer with multi-strategies coverage in wireless sensor networks. *Symmetry* 16(3), 286 (2024).
 19. Kang D W, Mo L P, Zhou K Q. An improved harmony search algorithm with segmented search. *IOP Conference Series: Materials Science and Engineering*. IOP Publishing 864(1), 012063 (2022).
 20. Khan A T, Li S, Cao X. Human guided cooperative robotic agents in smart home using beetle antennae search. *Science China Information Sciences* 65(2), 122204 (2022).
 21. Shang G X F, Huang Y R. Multi-objective optimization design of reluctance magnetic gear. *Journal of Magnetic Materials and Devices* 55(2),44-49 (2024).
 22. Sharma S., Saha, A.K., Chakraborty, S., et al. Quadratic and Lagrange interpolation-based butterfly optimization algorithm for numerical optimization and engineering design problem. *Soft Computing*, 29(1), 157-194 (2025).
 23. Sahoo Saroj Kumar et al. Self-adaptive moth flame optimizer combined with crossover operator and Fibonacci search strategy for covid-19 CT image segmentation. *Expert Systems with Application*, 227(Oct.), 120367.1-120367.34 (2023).
 24. Xue J, Shen B. A novel swarm intelligence optimization approach: sparrow search algorithm[J]. *Systems science & control engineering* 8(1), 22-34 (2020).
 25. Li N, Xue J K, SHU H S H. A sparrow search algorithm with adaptive t distribution mutation-based path planning of unmanned aerial vehicles. *Journal of Dong Hua University* 48(3), 69-74 (2022).
 26. Fu H, Liu H. Improved SSA based on multi-strategy fusion and its application. *Control and Decision* 37(01),87-96 (2022).
 27. Huang J Y. Research on sparrow search algorithm based on fusion of T distribution and Tent chaotic mapping. *Lanzhou University* 2021 (2022).
 28. Zhang W K, Liu S, Ren. Mixed strategy improved sparrow search algorithm. *Computer Engineering and Applications* 2021,74-82 (2021).
 29. Shi Y, Eberhart R. A modified particle swarm optimizer. 1998 IEEE international conference on evolutionary computation proceedings. IEEE world congress on computational intelligence. IEEE 1998, 69-73 (1998).
 30. Shi Y, Eberhart R C. Empirical study of particle swarm optimization. *Proceedings of the 1999 congress on evolutionary computation-CEC99*. IEEE 3, 1945-1950 (1999).
 31. Cai Y J, He X S, Yang X S. Salp swarm algorithm based on exponential inertial weights and adaptive mutation. *Basic Sciences Journal of Textile Universities* 34(2),108-116 (2021).
 32. Mao Q H, Zhang Q, Mao C C. Hybrid sine cosine algorithm and levy flying sparrow algorithm. *Journal of Shanxi University, Natural Science Edition* 44(06),1086-1091 (2021).
 33. Zhang Y, L Z G . Research on optimal configuration of photovoltaic dc microgrid based on levy flight Improved Bird Flock Algorithm. *Journal of Solar Energy* 42(05):214-220 (2021).
 34. Wolpert D H, Macready W G. No free lunch theorems for optimization. *IEEE transactions on evolutionary computation* 1(1), 67-82 (1997).
 35. Derrac J, García S, Molina D, Herrera F. A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm and Evolutionary Computation* 1(1), 3-18 (2011).
 36. Neri M, Campi A, Suffritti R, et al. SkyMedia-UAV-based capturing of HD/3D content with WSN augmentation for immersive media experiences. 2011 IEEE International Conference on Multimedia and Expo. IEEE, 2011, 1-6 (2011).
 37. Chang Chao, Xian Xiaodong, Hu Ying. Design of a remote environmental monitoring system for precision agriculture based on WSN. *Journal of Sensor Technology* 24(6): 5 (2011).

38. Nanda M, Kumar A, Kumar S. Localization of 3D WSN using Mamdani Sugano fuzzy weighted centroid approaches. IEEE, 2012 (2012).
39. Wang L, Qin N, Du X. WSN3D localization algorithm based on Tikhonov regularization method. Chinese Journal of Scientific Instrument 31(4),770-775 (2010).
40. Yao J Q, Tang B, Liu Z Y. Planning of distribution network in underdeveloped overseas cities based on improved k-means clustering algorithm. Electronic Measurement Technology 2021, 044-023 (2021).

ARTICLE IN PRESS