Computer Modeling in
Engineering & Sciences

Tech Science Press

DOI: 10.32604/cmes.2023.045096

ARTICLE

Research on Evacuation Path Planning Based on Improved Sparrow Search Algorithm

Xiaoge Wei[1,2,*], Yuming Zhang[1,2], Huaitao Song[1,2], Hengjie Qin[1,2] and Guanjun Zhao[3]

[1]The School of Building Environment Engineering, Zhengzhou University of Light Industry, Zhengzhou, 450001, China

[2]Zhengzhou Key Laboratory of Electric Power Fire Safety, Zhengzhou University of Light Industry, Zhengzhou, 450001, China

[3]Eccom Network System, Shanghai, 201103, China

*Corresponding Author: Xiaoge Wei. Email: 2015041@zzuli.edu.cn

Received: 17 August 2023 Accepted: 31 October 2023 Published: 29 January 2024

ABSTRACT

Reducing casualties and property losses through effective evacuation route planning has been a key focus for researchers in recent years. As part of this effort, an enhanced sparrow search algorithm (MSSA) was proposed. Firstly, the Golden Sine algorithm and a nonlinear weight factor optimization strategy were added in the discoverer position update stage of the SSA algorithm. Secondly, the Cauchy-Gaussian perturbation was applied to the optimal position of the SSA algorithm to improve its ability to jump out of local optima. Finally, the local search mechanism based on the mountain climbing method was incorporated into the local search stage of the SSA algorithm, improving its local search ability. To evaluate the effectiveness of the proposed algorithm, the Whale Algorithm, Gray Wolf Algorithm, Improved Gray Wolf Algorithm, Sparrow Search Algorithm, and MSSA Algorithm were employed to solve various test functions. The accuracy and convergence speed of each algorithm were then compared and analyzed. The results indicate that the MSSA algorithm has superior solving ability and stability compared to other algorithms. To further validate the enhanced algorithm's capabilities for path planning, evacuation experiments were conducted using different maps featuring various obstacle types. Additionally, a multi-exit evacuation scenario was constructed according to the actual building environment of a teaching building. Both the sparrow search algorithm and MSSA algorithm were employed in the simulation experiment for multi-exit evacuation path planning. The findings demonstrate that the MSSA algorithm outperforms the comparison algorithm, showcasing its greater advantages and higher application potential.

KEYWORDS

Sparrow search algorithm; optimization and improvement; function test set; evacuation path planning

1 Introduction

With the rapid urbanization of China and the emergence of densely populated places, the large-scale crowd gathering activities such as cultural tourism, conference forums, and sports events are constantly increasing. However, the frequency of safety accidents associated with these gatherings has surged significantly. For instance, a serious stampede accident occurred during the Halloween activities of the Itaewon in Seoul on October 29th, 2022, and the high-density crowd could not be effectively and timely evacuated, resulting in 156 deaths. In the early morning of March 1st, 2022, a fire broke out at the Ramad Shopping Center in the Syrian capital Damascus, causing 14 fatalities and 4 injuries. Consequently, the planning and design of evacuation routes for crowds in such emergencies has become a major focus of research [1].

One key component in the field of evacuation path research is the path planning algorithm. Based on their characteristics, these algorithms can be categorized into classical intelligent optimization algorithms and heuristic intelligent optimization algorithms [2]. Classical intelligent optimization algorithms include algorithms such as the A∗ algorithm [3,4] and Tabu search algorithm [5,6], while heuristic intelligent optimization algorithms encompass methods like grey wolf optimizer (GWO) [7], ant optimization algorithm [8], particle swarm optimization [9], etc. Additionally, machine learning-based algorithms have also been used by a large number of scholars in the study of path planning, such as neural networks [10], support vector machines, or reinforcement learning techniques. Compared with the dynamic path planning process of evacuation, classical intelligent optimization algorithms suffer from drawbacks such as poor real-time performance and high randomness, and they are no longer able to meet the needs of evacuation path planning. The heuristic intelligent optimization algorithms can partially overcome these limitations. Thus, heuristic intelligent optimization algorithms have been extensively studied by researchers due to their ability to provide fast solving speed, high accuracy, and excellent real-time performance. Many scholars have designed and proposed numerous intelligent optimization algorithms [11,12]. The intelligent optimization algorithms that excel in path planning include the ant colony optimization algorithm inspired by the foraging process of the ant colony [13], the fruit fly optimization algorithm proposed by simulating the predatory behavior of fruit flies using their keen sense of smell and vision [14], the grey wolf optimization algorithm proposed based on the internal social hierarchy and hunting behavior of the grey wolf population [15], and the whale optimization algorithm proposed by imitating the social behavior of humpback whales [16].

Inspired by the foraging and anti-predatory behavior of sparrows, Xue et al. designed the sparrow search algorithm (SSA) in 2020 [17]. Compared to other biomimetic intelligence algorithms, the sparrow search algorithm exhibits strong problem-solving ability and faster convergence. As a result, the sparrow search algorithm has attracted the attention of scholars and has been quickly applied to different research fields, including power load forecasting [18], image processing [19], robot path tracking [20], sensor network performance optimization [21], wireless positioning [22] and fault diagnosis [23]. However, the basic sparrow search algorithm has low convergence accuracy in solving multi-drone collaborative trajectory planning problems and is prone to falling into local optima. Zhang et al. [24] proposed a modified sparrow algorithm using a logarithmic spiral strategy and an adaptive ladder strategy. This adaptation allows it to plan coordinated flight trajectories with nearly optimal cost and constraint conditions while ensuring time coordination. Jiang et al. [25] transformed the route planning problem into a multidimensional function optimization problem by establishing a three-dimensional task space model and a cost function for unmanned aerial vehicle route planning. Additionally, they incorporated a chaos strategy, leveraging the convergence speed and exploration ability of an adaptive inertia weight balancing algorithm. Furthermore, a trajectory planning method based on a chaotic sparrow search algorithm was proposed by others [26] to solve the problems of high computational complexity and difficult convergence in drone trajectory planning.

Given the widespread application of the sparrow search algorithm in unmanned aerial vehicle trajectory planning, we aim to utilize this algorithm for personnel evacuation path planning. However, due to the different goals and application scenarios between the two types of application areas, the factors that the path algorithm needs to consider are also different. Drone trajectory planning [27] involves determining an appropriate path for a drone to execute specific tasks (e.g., material transportation, patrols, etc.) while ensuring the drone reaches the target location with precision, efficiency, and safety. Factors considered in trajectory planning include the position of the drone, flight speed, flight altitude, terrain, wind conditions, obstacles, etc. [28]. This planning is usually completed by robot control algorithms and can be achieved through mathematical models and optimization algorithms. Pedestrian evacuation path planning [29] refers to the planning of a suitable path to respond to emergencies, enabling pedestrians to evacuate quickly and safely to secure areas. The factors that need to be considered in path planning include personnel distribution, safety exits, path restrictions such as stairs, as well as personnel flow, and evacuation time [30].

The sparrow search algorithm can plan routes based on the current location, target location, obstacles and other information. It then optimizes decisions within the stochastic process using available information. However, the sparrow algorithm also possesses certain flaws and shortcomings. In the early stages of algorithm iteration, the search range is limited, and cannot perform effective global searches. And there is a risk of falling into local optimal solutions in the later stage of iteration. Therefore, there is scope for targeted optimization [31]. In response to these problems, scholars continue to optimize and improve this algorithm. For instance, Xin et al. [32] initialized the population through a Tent chaotic sequence. This strategy increased the diversity of the initial population and enhanced the global search capability of the algorithm. Wei et al. [33] employed the logistic-tent mapping for population initialization to enhance diversity in the population. Also, an adaptive period factor is introduced into the producer's update position equation. This manuscript addresses the deficiencies of the aforementioned sparrow search algorithm and introduces an improved sparrow search algorithm (MSSA). The major contributions are summarized as follows:

(1) The Golden Sine Cosine Algorithm and a nonlinear weight factor are introduced into the Sparrow Algorithm Producer Update Equation, thereby enhancing the global and local search capabilities of the discoverers.

(2) The Gaussian Cauchy perturbation strategy is added to the proposed algorithm, which is to improve its ability to escape local optima.

(3) The strategy of mountain climbing is presented to improve the local search ability of the SSA algorithm.

To evaluate the effectiveness of the improved algorithm, we conducted comparative experiments using five different algorithms, including the SSA algorithm, MSSA algorithm, WOA algorithm, GWO algorithm, and IGWO algorithm, to solve the standard test functions, aiming to verify and assess the effectiveness of the improved algorithm. Ultimately, the comparative experiments with the evacuation simulation model built by the grid method were carried out to prove the proposed algorithm's potential for practical engineering applications.

2 Basic Principles of Sparrow Search Algorithm

The sparrow search algorithm [17] initially divides the sparrow population into two different types: discoverers and followers. Then the third type of individuals, i.e., the scouts, is randomly chosen among the discoverers and followers. The discoverers are responsible for locating food sources for the entire population. During the food search, the followers may temporarily deviate from the following state and explore other places in search of abundant food sources. On the other hand, the scouts are responsible for issuing danger alarm signals to alert the population.

The sparrows follow the discoverers, and the number of discoverers accounts for 10%–20% of the total population. The equation for updating the position of the discoverers in the sparrow search algorithm is shown in Eq. (1).

$$x_{i,j}^{t+1} = \begin{cases} x_{i,j}^{t} \cdot \exp\left(-\dfrac{i}{\alpha \cdot iter_{\max}}\right), & if \;\; R_2 < ST \\ x_{i,j}^{t} + Q \cdot L \qquad\quad , & if \;\; R_2 \geq ST \end{cases} \tag{1}$$

In Eq. (1), $x_{i,j}^{t}$ represents the position of the i-th sparrow individual in the j-th dimension of the t-th cycle. α is a random number with a value range of [0, 1]. $iter_{max}$ denotes the maximum number of iterations. R_2 is the warning value, and ST represents the safety value. When $R_2 < ST$, the discoverer can search within the current location range. When $R_2 \geq ST$, the discoverer will narrow the search range and lead the sparrow population to move to other areas. Q is a random number subject to normal distribution. L is a matrix of 1 × d with all elements equal to 1. This matrix can be omitted for the position information of each sparrow individual, and the calculation process will be simplified for each dimension of the sparrow individual to reduce the overall calculation amount and computational complexity. The equation for updating the discoverer's position by omitting matrix L is shown in Eq. (2).

$$x_{i,j}^{t+1} = \begin{cases} x_{i,j}^{t} \cdot \exp\left(-\frac{i}{\alpha \cdot iter\max}\right), & if \;\; R_2 < ST \\ x_{i,j}^{t} + Q \qquad\qquad , & if \;\; R_2 \geq ST \end{cases} \tag{2}$$

Within the iteration cycle, the discoverer is the sparrow with the best position among individuals in the sparrow population, while the other sparrows act as followers and scouts. The equation for updating the positions of the followers is shown in Eq. (3).

$$x_{i,j}^{t+1} = \begin{cases} Q \cdot \exp\left(\frac{xw_{i,d}^{t} - x_{i,d}^{t}}{i^2}\right) \qquad\qquad , & if \;\; i > n/2 \\ xb_{i,d}^{t} + \left|x_{i,d}^{t} - x_{p}^{t+1}\right| \cdot A^{+} \cdot L, & otherwise \end{cases} \tag{3}$$

In Eq. (3), Q is a random number that follows a normal distribution, A is a matrix of 1 ∗ D. L is also a matrix of 1 ∗ d dimensions and all elements are 1. Both A and L can be omitted, and the equation of followers position update optimized as shown in Eq. (4).

$$x_{i,d}^{t+1} = \begin{cases} Q \cdot \exp\left(\frac{xw_{i,d}^{t} - x_{i,d}^{t}}{i^2}\right) \qquad\qquad\qquad\qquad\qquad , & if \;\; i > n/2 \\ xb_{i,d}^{t} + \frac{1}{D}\sum\limits_{D=1}^{D}\left(rand\{-1,1\} \cdot \left(\left|xb_{i,d}^{t} - x_{i,d}^{t}\right|\right)\right), & if \;\; i \leq n/2 \end{cases} \tag{4}$$

In Eq. (4), $xw_{i,d}^{t}$ is the worst-case position of the i-th sparrow in the d-th dimension during the t-th iteration. $xb_{i,d}^{t}$ denotes the optimal sparrow position within the population during the t-th iteration. $x_{i,d}^{t}$ represents the optimal position of the i-th sparrow in the d-th dimension during the t-th iteration of the population. When $i > n/2$, the fitness value of this follower is poor, and the follower will move to a position with better fitness. When $i \leq n/2$, it indicates that the individual fitness value of this follower is good, approaching the finder's position, and searching for food could be carried out near this position.

Within the sparrow population, 10% to 20% of individuals are responsible for reconnaissance and warning. In each iteration, sparrow individuals are randomly selected from discoverers and followers to become scouts. The equation for updating the scout's position is shown in Eq. (5).

$$x_{i,d}^{t+1} = \begin{cases} xb_{i,d}^{t} + \beta \cdot \left(x_{i,d}^{t} - xb_{i,d}^{t}\right), & f_i \neq f_g \\ x_{i,d}^{t} + k \cdot \left(\frac{x_{i,d}^{t} - xw_{i,d}^{t}}{\left|f_i - f_w\right| + \varepsilon}\right) \quad , & f_i = f_g \end{cases} \tag{5}$$

In Eq. (5), $xb_{i,d}^{t}$ represents the optimal sparrow position within the population during the t-th iteration. β is a random number subject to standard normal distribution. k is a uniform random number with a value range of $[-1, 1]$. ε is a minimum constant that prevents the denominator from being unique. f_i is the fitness value of the i-th sparrow individual. f_w is the optimal fitness value in the current iteration. f_w is the worst fitness value in the current iteration. When $f_i = f_g$, the sparrow at the best position in the population will move to a position nearby as a scout. When $f_i \neq f_g$, this scout sparrow individual is not in the optimal position within the population and needs to move towards the optimal position in the population.

3 Improved Sparrow Search Algorithm

3.1 Adding Golden Sine Algorithm to Improve Update Strategy

The Golden Sine Algorithm, proposed by Tanyildizi et al. in 2017, is based on scanning within the unit circle of the sine function, similar to the spatial search for target solving [34]. This algorithm explores all points within the unit circle and integrates the golden section method to approach the optimal solution. The algorithm has the characteristics of fewer parameter settings, good robustness, and fast convergence speed.

The algorithm first randomly generates S population individual positions, corresponding to the potential solutions of the target problem. The equation for updating the position of the i-th individual is shown in Eq. (6).

$$X_i^d(t+1) = X_i^d(t) * |\sin(r_1) + r_2 * \sin(r_1)| * X_1 * p^d(t) - X_2 * X_i^d(t) \tag{6}$$

In Eq. (6), $X_i^d(t)$ represents the position of the i-th individual in the t-th iteration of the d-dimensional space. $P^d(t)$ is the global optimal position of the individual in the t-th iteration. r_1 is a randomly generated number within the range of $[0, 2\pi]$, which affects the individual's movement distance. r_2 is a random number within the range of $[0, \pi]$, which affects the direction of individual position updates. X_1 and X_2 are coefficients of the golden section number, and the golden section number τ is an irrational number defined as $\left(\sqrt{5-1}/2\right)$. The introduction of these coefficients enables the algorithm to converge quickly while approaching the optimal value. The expressions of X_1 and X_2 are shown in Eq. (7).

$$\begin{aligned} X_1 &= \alpha\tau + b(1-\tau) \\ X_2 &= \alpha(1-\tau) + b\tau \end{aligned} \tag{7}$$

In Eq. (7), the initial values of α and b are set to be $-\pi$ and π, and then the values of α and b change with the target value. The values of X_1 and X_2 are also updated accordingly.

Based on the description of the Golden Sine algorithm, it can be seen that during the iteration process, the individual optimal position $X_i^d(t)$ and global optimal position $P^d(t)$ have a good fit with the individual optimal position and global optimal position of the sparrow population. Hence, in the code design of the algorithm, the golden sine algorithm can be introduced as two factors to replace the sparrow search algorithm to improve the global and local search capabilities of the sparrow population discoverer.

3.2 Nonlinear Weight Factor Optimization Strategy

On the basis of adding the golden sine algorithm, the algorithm further optimizes the location update method of the discoverers by introducing a nonlinear weight factor ω_2. The calculation equation is provided in Eq. (8).

$$\omega_2 = 1 - \left(\cos\left(t * \frac{pi}{Max\_iter} + \omega_{\max}\right) * \frac{(\omega_{\max} + \omega_{\min})}{2} + \alpha\right) \tag{8}$$

In Eq. (8), t represents the current number of iterations, and *pi is the* π.*Max_iter* denotes the maximum number of iterations. $\omega_{\max}$, $\omega_{\min}$ and α are the calculation parameters for the weight factor. $\omega_{\max} = 0.9$, $\omega_{\min} = 0.2$, $\alpha = 0.45$. The MSSA algorithm, incorporating the optimization strategies of the Golden Sine algorithm and the nonlinear weight factor, updates the discoverer position code segment as depicted in Table 1.

Table 1: Introduction of Golden Sine Algorithm optimization strategy and nonlinear weight factor

Code
if $R_2 < ST$
for i = 1:recover_Num
X(Index(i),:) = ω_2 * pbest(Index(i),:) * abs(sin(r_1))-
r_2 * sin(r_1) * abs(c_1 * gbest-c_2 * pbest(Index(i),:));
% Discoverer location update
fit(Index(i)) = fun(X(Index(i),:),G);   % Update of fitness value
end
else
X(Index(i),:) = pbest(Index(i),:) + randn(1) * ones(1, dim); % Fly to other places
end

3.3 Fusion of Mountain Climbing Method and Gaussian-Cauchy Perturbation Strategy

Heuristic algorithms can effectively reduce the solving time and improve the operational efficiency of the algorithm. In this article, we propose the MSSA algorithm introduces the mountain climbing method as a local search strategy. Then the search accuracy of the individual optimal position and optimal fitness value of the sparrow population could be improved. When compared with the simulated annealing algorithm and tabu search algorithm, the mountain climbing method has the advantage of reducing algorithm runtime and improving algorithm efficiency. Incorporating the mountain climbing method as an optimization strategy for the optimal position generated by the iterative cycle can quickly traverse the area near the optimal position searched by the algorithm. However, the mountain climbing method may become trapped in local optima. Therefore, it is necessary to increase the ability to jump out of the current position while traversing the surrounding positions. To address this, a Gaussian-Cauchy perturbation strategy is incorporated into the optimal position. The implementation logic of this strategy is to increase the perturbation coefficient to the search range of the current loop optimal position when the mountain climbing method greedily traverses it. This perturbation coefficient can offer a probability of escaping the current optimal position and enhancing the search ability of the global optimal position.

The MSSA algorithm proposed in this article adopts a Gaussian-Cauchy perturbation strategy, which can combine these two types of perturbations. Before calculating the perturbation of the optimal position, a random number is generated. If the random number is greater than 0.5, Gaussian perturbation will be used for calculation. Conversely, if the random number is less than 0.5, Cauchy perturbation will be used. The obtained perturbation probability will participate in the greedy traversal search for the optimal position of the mountain climbing method. Both distributions can improve and enhance the algorithm's capability to escape local optima when utilized for algorithm optimization. The code snippet for the Gaussian-Cauchy perturbation strategy is shown in Table 2.

Table 2: The code snippet for Gaussian-Cauchy perturbation strategy

Code	
if rand > 0.5	
bestX_rct = gbest.*(1 + gauss);	% Gaussian perturbation strategy
else	
bestX_rct = gbest.*(1 + cauchy);	% Cauchy perturbation strategy
end	

bestX_rct represents the global most significant position after adding the disturbance strategy, which will be involved in the calculation of greedy traversal of the mountain climbing method in the future. gbest is the global optimal position within the current iteration cycle.

3.4 The Running Process of the MSSA Algorithm

This section combines the above four improvements with traditional SSA algorithms, including the Golden Sine Cosine Algorithm optimization strategy, nonlinear weight factor optimization strategy, mountain climbing local search strategy, and Gaussian Cauchy perturbation strategy. Then, an improved Sparrow Optimization Algorithm (MSSA) was proposed. Referring to the flowchart of the algorithms in the literature [35], the flowchart of MSSA is shown in Fig. 1.

3.5 Time Complexity Analysis of MSSA Algorithm

To evaluate the performance of the improved MSSA algorithm, the time complexity analysis was conducted. In this analysis, N is the total number of the sparrow population, and n represents the individual dimension. t_0 represents the initialization time, while t_1 is the initialization time of each dimension of the sparrow population. The calculation time for the individual fitness value of the sparrow population is denoted by $f(n)$, and the time for determining the distribution of discoverers and followers in sparrow population based on the fitness value is t_2. The time complexity of the MSSA algorithm and SSA algorithm during the initialization stage is the same, as shown in Eq. (9).

$$T_1 = o(t_0 + N \cdot (n \cdot t_1 + f(n) + t_2) = o(n + f(n)) \tag{9}$$

After initialization, the algorithm enters the iteration loop stage, with *Max_iter* representing the maximum number of iterations. Within a single iteration cycle, the time required for calculating the nonlinear weight factor ω_2 is d_1. As the MSSA algorithm is improved by the Golden Sine algorithm, which directly updates the location of the discoverer without any additional computational steps, the time required for updating the location of the discoverer in the MSSA algorithm is t_3. The time required for updating the follower positions of the sparrow population is t_4. The time required for updating the position of sparrow individuals responsible for detecting and warning sparrow populations is t_5. The time required to calculate the individual position of the sparrow population is t_6. The time required to calculate the disturbance probability based on the Gaussian Cauchy perturbation strategy is d_2. The time required to calculate the optimal position of the sparrow population using the local search strategy mountain climbing method is d_3. If the time to calculate the individual fitness value of the sparrow population is $f(n)$, the time complexity of the sparrow search algorithm within a single iteration cycle is shown as Eq. (10):

$$T_2 = o(N \cdot (d_1 + t_3 + t_4 + t_5) + n \cdot (t_6 + d_2 + d_3) + f(n)) = o(n + f(n)) \tag{10}$$

Since *Max_iter* is a fixed value, the time complexity of the iteration cycle of the MSSA algorithm can be considered as the product of the time complexity of a single iteration cycle and the total number of iterations, then the time complexity of the sparrow search algorithm is given by Eq. (11):

$$T = T_1 + Max\_iter \cdot T_2 = o(n + f(n)) \tag{11}$$

Based on the above analysis, it can be concluded that the time complexity of the SSA algorithm and MSSA algorithm is of the same order of magnitude. MSSA algorithm not only enhances global search and local search capabilities, but also does not affect the actual operational efficiency of the algorithm.

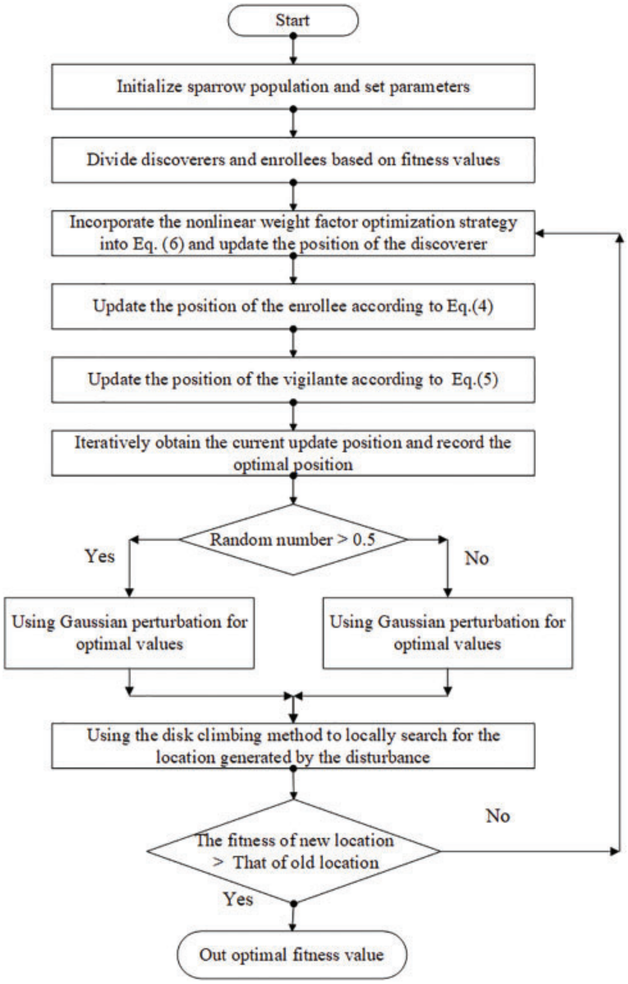

Start
Initialize sparrow population and set parameters
Divide discoverers and enrollees based on fitness values
Incorporate the nonlinear weight factor optimization strategy into Eq. (6) and update the position of the discoverer
Update the position of the enrollee according to Eq.(4)
Update the position of the vigilante according to Eq.(5)
Iteratively obtain the current update position and record the optimal position
Random number > 0.5
Yes
No
Using Gaussian perturbation for optimal values
Using Gaussian perturbation for optimal values
Using the disk climbing method to locally search for the location generated by the disturbance
No
The fitness of new location > That of old location
Yes
Out optimal fitness value

Figure 1: MSSA algorithm flowchart

4 Optimization Achievement Test of Improved Sparrow Search Algorithm

The performance testing of intelligent optimization algorithms requires the application of the standard test functions and functionalized engineering problems. Speed and robustness should be assessed by comparing the results. The specific steps involve determining the performance of different intelligent optimization algorithms by comparing their optimal values, average values, and variance using standard test functions under the same experimental environment. Alternatively, by examining the optimal solution change curve of the algorithm during the iteration process, we could analyze which algorithm exhibits a faster descent speed from the origin to any number of iterations to assess their convergence speed. The above performance testing methods mainly evaluate the performance of intelligent optimization algorithms from two perspectives: computer resource consumption and the quality of results obtained. These criteria are vital for evaluating algorithm performance.

4.1 Introduction to Standard Function Test Set

In response to the MSSA algorithm proposed in Section 3, this chapter employs the standard test functions to test its optimization and solving capabilities, where the functions F_1~F_5 are the unimodal test functions, and the functions F_6~F_{10} represent multimodal testing functions. Additionally, the whale algorithm (WOA), grey wolf algorithm (GWO), and improved grey wolf algorithm (IGWO) integrating CS algorithm strategy are introduced for comparative testing of optimization capabilities. The test results are analyzed based on two criteria: optimization accuracy and convergence speed, which are aimed at verifying the effectiveness of the improved optimization strategy. To ensure accuracy, reliability and mitigate the impact of sporadic results on the evaluation of algorithm performance, the five algorithms participating in comparative analysis will undergo 50 repeated experiments. These experiments will analyze the optimal value, average value, and variance of the results. The optimal value of the search results reflects the upper limit of the search ability of each algorithm, i.e., the ability to approximate the optimal value of the test function. The average value represents the central value within the set of multiple search results, thereby avoiding the impact of accidental optimal results on the assessment of the algorithm's search ability. By calculating the variance of the solution results, it is possible to visually assess the degree of dispersion in the distribution of each algorithm's solution results, thereby reflecting the advantages and disadvantages of each algorithm. In addition, the simulation experiment software and hardware platforms used are consistent. The operating system is Windows 10, and the simulation software employed MATLAB R2018b. Table 3 illustrates the 10 standard test functions used in this article.

Table 3: The standard test function

Number	Function	Function dimension N	Variable value range S	Optimal value F
	Unimodal test function			
F_1	Sphere Function	30/60/90	$[-100;100]^n$	0
F_2	Schwefel's Problem 2.22	30/60/90	$[-10;10]^n$	0
F_3	Schwefel's Problem 1.2	30/60/90	$[-100;100]^n$	0
F_4	Schwefel's Problem 2.21	30/60/90	$[-100;100]^n$	0

(Continued)

Table 3 (continued)

Number	Function	Function dimension N	Variable value range S	Optimal value F
F_5	Generalized Rosenbrock's Function	30/60/90	$[-30;30]^n$	0
	Multimodal test function			
F_6	Generalized Schwefel's Problem 2.26	30/60/90	$[-500;500]^n$	−12569.5
F_7	Generalized Rastrigin's Function	30/60/90	$[-5.12;5.12]^n$	0
F_8	Ackley's Function	30/60/90	$[-32;32]^n$	0
F_9	Generalized Griewank's Function	30/60/90	$[-600;600]^n$	0
F_{10}	Generalized Penalized Function 1	30/60/90	$[-50;50]^n$	0

4.2 Comparative Analysis of Optimization Results

Table 4 displays the results obtained by solving unimodal test functions for five algorithms. A detailed comparison is made among the optimal value, mean value, and variance to further evaluate the solving ability of the improved algorithm.

Table 4: The results of solving the test function

Function	Algorithm	d = 30			d = 60			d = 90		
		Optimal	Mean	Variance	Optimal	Mean	Variance	Optimal	Mean	Variance
F_1	SSA	1.25E − 18	8.72E − 66	7.61E − 13	0.00E + 00	2.09E − 18	4.35E − 35	4.23E − 15	1.14E − 68	1.31E − 135
	MSSA	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	8.17E − 97	6.01E − 192	0.00E + 00	0.00E + 00	0.00E + 00
	WOA	5.19E − 72	8.11E − 62	6.57E − 12	2.34E − 71	7.74E − 66	4.96E − 130	5.22E − 71	1.27E − 58	1.61E − 115
	GWO	3.38E − 29	8.42E − 28	6.03E − 55	0.00E + 00	1.13E − 17	8.59E − 35	5.42E − 14	2.11E − 13	2.27E − 26
	IGWO	8.30E − 35	3.17E − 33	2.23E − 65	0.00E + 00	6.57E − 22	1.31E − 42	1.07E − 17	6.86E − 17	2.55E − 33
F_2	SSA	6.60E − 67	6.94E − 37	3.00E − 72	1.63E − 79	3.16E − 34	9.07E − 67	2.55E − 73	1.03E − 34	1.06E − 67
	MSSA	3.29E − 18	1.99E − 16	0.00E + 00	1.49E − 18	9.72E − 16	0.00E + 00	1.02E − 18	1.17E − 16	0.00E + 00
	WOA	5.21E − 54	8.62E − 48	6.35E − 94	1.43E − 54	3.28E − 47	5.94E − 93	1.89E − 52	4.44E − 44	8.65E − 87
	GWO	8.94E − 18	9.66E − 17	4.06E − 33	2.25E − 11	3.55E − 11	8.18E − 23	7.96E − 09	1.18E − 08	2.35E − 17
	IGWO	2.03E − 21	1.96E − 20	4.23E − 40	5.77E − 14	1.55E − 13	6.33E − 27	4.39E − 11	7.89E − 11	1.63E − 21
F_3	SSA	1.25E − 12	3.12E − 54	9.74E − 11	2.64E − 28	1.34E − 56	1.75E − 111	1.81E − 23	4.36E − 64	1.88E − 126
	MSSA	1.10E − 30	1.88E − 26	0.00E + 00	1.35E − 30	1.65E − 26	0.00E + 00	2.13E − 29	7.19E − 23	0.00E + 00
	WOA	1.78E + 04	5.21E + 04	2.12E + 08	2.40E + 05	3.77E + 05	7.60E + 09	2.66E + 05	4.10E + 05	1.91E + 10
	GWO	1.67E − 07	1.03E − 05	6.19E − 10	1.75E − 01	3.64E + 00	1.63E + 01	1.97E + 01	2.01E + 02	2.95E + 04
	IGWO	9.74E − 10	2.48E − 07	3.65E − 13	3.73E − 04	3.35E − 01	6.63E − 01	1.58E − 01	1.45E + 02	2.45E + 04
F_4	SSA	3.34E − 76	4.36E − 37	1.65E − 72	1.75E − 11	2.34E − 40	5.49E − 79	7.14E − 10	2.35E − 41	5.55E − 81
	MSSA	4.37E − 20	5.54E − 17	0.00E + 00	4.19E − 30	5.78E − 17	0.00E + 00	3.00E − 19	3.48E − 16	0.00E + 00
	WOA	1.77E + 01	5.76E + 01	4.52E + 02	6.42E + 01	8.51E + 01	7.61E + 01	1.12E + 01	6.19E + 01	1.11E + 03
	GWO	9.56E − 08	9.87E − 07	1.17E − 12	3.73E − 04	2.75E − 03	1.02E − 05	4.15E − 02	3.62E − 01	4.47E − 01
	IGWO	6.92E − 10	5.76E − 09	3.33E − 17	1.55E − 05	5.28E − 05	4.78E − 10	1.30E − 03	5.46E − 03	4.68E − 05
F_5	SSA	7.12E − 07	5.51E − 05	6.75E − 09	9.01E − 05	2.61E − 04	2.34E − 08	5.00E − 08	1.79E − 04	7.41E − 08
	MSSA	3.11E − 09	4.28E − 05	3.94E − 09	8.10E − 07	3.71E − 04	5.69E − 07	1.15E − 06	4.61E − 04	1.12E − 06
	WOA	2.77E + 01	2.82E + 01	2.00E − 01	5.77E + 01	5.84E + 01	7.78E − 02	8.82E + 01	8.84E + 01	1.59E − 02

(Continued)

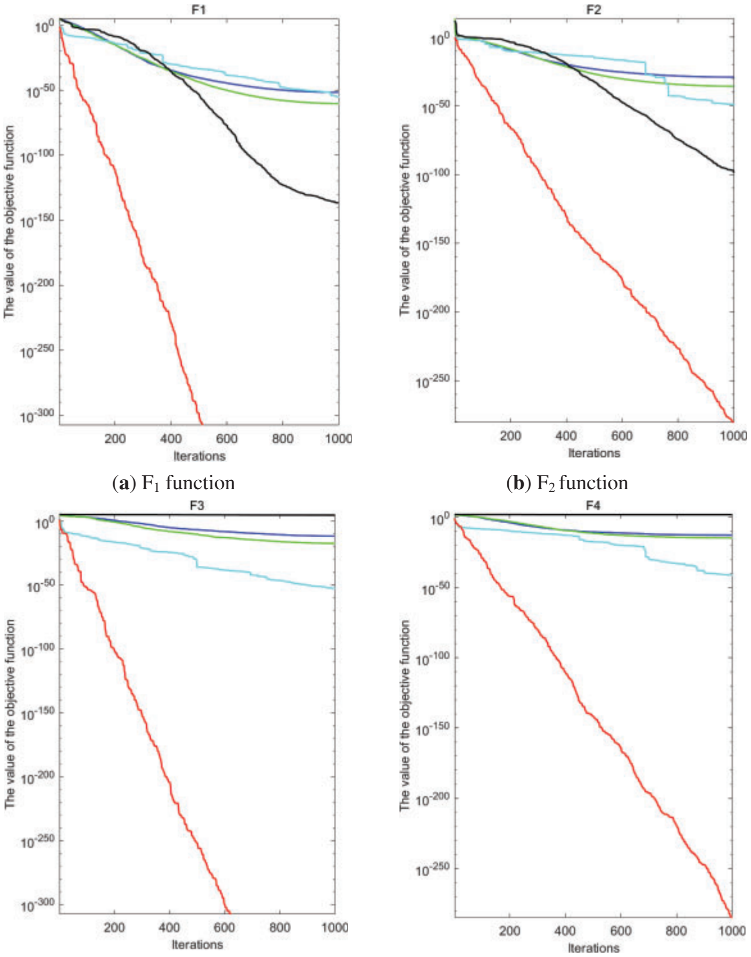
Table 4 (continued)

Function	Algorithm	d = 30			d = 60			d = 90		
		Optimal	Mean	Variance	Optimal	Mean	Variance	Optimal	Mean	Variance
	GWO	2.56E + 01	2.69E + 01	7.48E − 01	5.70E + 01	5.79E + 01	3.89E − 01	8.60E + 01	8.76E + 01	5.52E − 01
	IGWO	2.63E + 01	2.71E + 01	3.22E − 01	5.69E + 01	5.77E + 01	5.02E − 01	8.61E + 01	8.80E + 01	6.51E − 01
F_6	SSA	−8.97E + 03	−7.90E + 03	4.36E + 05	−1.58E + 04	−1.48E + 04	7.21E + 05	−2.37E + 04	−2.17E + 04	2.23E + 06
	MSSA	−1.26E + 04	−1.20E + 04	6.13E + 05	−1.50E + 04	−1.50E + 04	1.47E − 23	−3.77E + 04	−3.57E + 04	6.66E + 05
	WOA	−1.25E + 04	−9.83E + 03	2.47E + 06	−2.51E + 04	−2.17E + 04	1.55E + 07	−3.64E + 04	−2.77E + 04	1.20E + 07
	GWO	−6.97E + 03	−6.17E + 03	4.28E + 05	−1.26E + 04	−9.72E + 03	4.58E + 06	−1.78E + 04	−1.56E + 04	2.33E + 06
	IGWO	−5.07E + 03	−4.43E + 03	1.71E + 05	1.12E + 04	−8.04E + 03	4.99E + 06	−1.48E + 04	−1.16E + 04	2.96E + 06
F_7	SSA	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00
	MSSA	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00
	WOA	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00
	GWO	5.68E − 14	7.44E − 01	4.07E + 00	1.59E − 12	3.97E + 00	1.59E + 01	8.29E − 11	3.56E + 00	1.44E + 01
	IGWO	0.00E + 00	8.19E − 13	2.03E − 24	2.27E − 13	1.23E + 00	8.10E + 00	1.02E − 12	4.03E − 01	1.62E + 00
F_8	SSA	8.88E − 16	3.73E − 15	8.08E − 29	8.88E − 16	8.88E − 16	4.32E − 62	8.88E − 16	8.88E − 16	4.32E − 62
	MSSA	8.88E − 16	8.88E − 16	4.32E − 62	8.88E − 16	8.88E − 16	4.32E − 62	8.88E − 16	8.88E − 16	4.32E − 62
	WOA	8.88E − 16	4.09E − 15	4.07E − 30	0.00E + 00	4.35E − 15	9.19E − 30	8.88E − 16	4.09E − 15	4.07E − 30
	GWO	7.90E − 14	9.82E − 14	1.04E − 28	1.84E − 10	4.70E − 10	5.96E − 20	8.33E − 10	5.75E − 08	1.69E − 15
	IGWO	2.93E − 14	3.39E − 14	2.26E − 29	1.15E − 12	3.34E − 12	3.33E − 24	6.32E − 10	9.90E − 10	2.79E − 19
F_9	SSA	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00
	MSSA	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00
	WOA	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00
	GWO	0.00E + 00	1.12E − 02	1.39E − 04	0.00E + 00	7.81E − 03	1.67E − 04	3.89E − 14	2.05E − 03	4.20E − 05
	IGWO	0.00E + 00	1.54E − 03	2.38E − 05	0.00E + 00	2.31E − 03	5.32E − 05	0.00E + 00	3.73E − 03	6.20E − 05
F_{10}	SSA	1.01E − 11	1.42E − 03	2.01E − 05	5.43E − 10	9.48E − 09	1.26E − 16	7.82E − 10	5.57E − 08	1.73E − 14
	MSSA	7.20E − 11	4.03E − 09	4.93E − 17	4.13E − 12	4.95E − 08	1.05E − 14	6.50E − 13	2.18E − 09	2.59E − 15
	WOA	3.42E − 02	1.14E − 01	3.30E − 02	0.00E + 00	4.35E − 15	9.19E − 30	4.70E − 02	9.15E − 02	2.09E − 03
	GWO	2.08E − 02	5.26E − 02	8.39E − 04	7.72E − 02	1.42E − 01	2.65E − 03	1.27E − 01	2.21E − 01	3.28E − 03
	IGWO	1.08E − 02	3.34E − 02	2.84E − 04	5.21E − 02	1.02E − 01	1.71E − 03	1.72E − 01	2.25E − 01	1.71E03

To analyze the performance of the MSSA algorithm more intuitively, we compared the results (including optimal value, mean value, and variance) under identical conditions. Table 4 reveals that, except for solving functions 6 and 5, the MSSA algorithm has slightly worse computational performance compared to some algorithms, while in other cases it generally shows strong computational power. The results of solving the different functions under 30-dimensional, 60-dimensional, and 90-dimensional conditions show that although the MSSA algorithm does not outperform other algorithms under all conditions, it has significant advantages overall compared to other algorithms in the optimal value, mean value, and variance. The results prove that the MSSA algorithm possesses strong computational capabilities for solving problems, thereby establishing a solid foundation for subsequent practical engineering applications.

4.3 Comparative Analysis of Convergence Curve Results of Test Functions

Fig. 2 illustrates the convergence curves for the iteration of different functions under 30-dimensional conditions. The dark blue line, green line, sky blue line, red line, and black line in Fig. 2 represent the calculation results of GWO, IGWO, SSA, MSSA, and WOA, respectively. The convergence curves of the functions $F_1 \sim F_{10}$ intuitively illustrate that the MSSA algorithm exhibits rapid convergence speed and can quickly approach the optimal value of the function during the early stages of convergence, which has a significant advantage compared to other algorithms. By conducting a detailed analysis of the convergence curves, it can be concluded that the MSSA algorithm has a significant advantage in convergence speed during solving the functions F_1, F_2, F_3, F_4, F_6, F_8, F_9, and F_{10}. While solving the function F_5, the SSA algorithm initially exhibits slightly faster convergence speed than the MSSA algorithm. However, it becomes evident that the MSSA algorithm possesses a stronger ability to escape local optima and converge towards the optimal value faster after 200 iterations. Moreover, when solving the function F_7, there is no significant difference in convergence speed among the five algorithms during the initial iteration stage. However, in the middle of the iteration, all five algorithms fall into local optima, and the convergence curves become horizontal. In the later stage of the iteration, the MSSA algorithm outperforms others by successfully escaping local optima and approaching the optimal value, hereby exhibiting distinct advantages over other comparative algorithms.



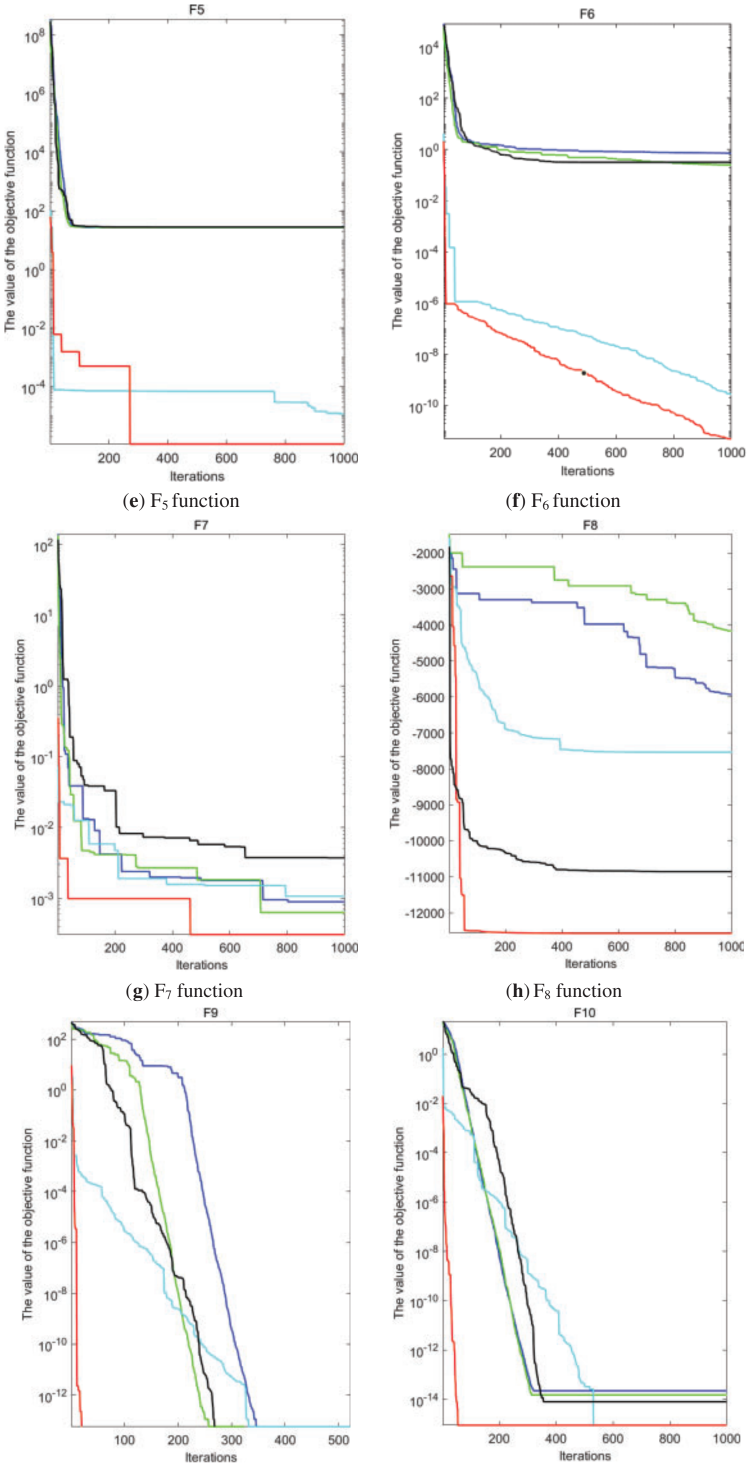
(**a**) F_1 function

(**b**) F_2 function

(**c**) F_3 function

(**d**)F_4 function

Figure 2: (Continued)



(**e**) F$_5$ function

(**f**) F$_6$ function

(**g**) F$_7$ function

(**h**) F$_8$ function

(**i**) F$_9$ function

(**j**) F$_{10}$ function

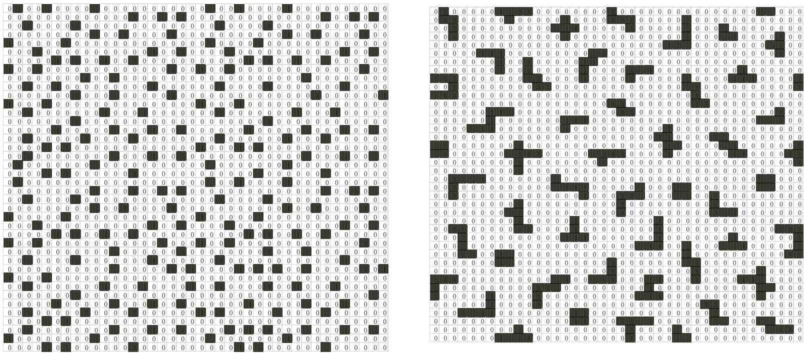
Figure 2: Convergence curves of different algorithms for solving different test functions

5 Application of the MSSA Algorithm in Path Planning

Based on the achievement test of the improved algorithm in the fourth section, the path planning capability of the algorithm in evacuation scenarios is evaluated. Initially, the grid method is employed to construct evacuation path planning scenes with different types of obstacles. Subsequently, the five algorithms are utilized for path planning experiments, and the path results are compared and analyzed. Additionally, based on the actual teaching building environment, a simplified grid map is created. Consequently, the MSSA algorithm is then applied in multi-exit evacuation path planning simulation experiments to verify the improved algorithm's evacuation path planning efficacy in multi-exit environments.

5.1 Evacuation Environment Models with Different Obstacle Distribution

There are two types of map types set up in the simulation experiment. The obstacle layout of map 1 mainly considers the pathfinding ability of the test algorithm under the scattered distribution of obstacles, while the obstacle layout of map 2 mainly considers the pathfinding ability of the test algorithm in multi-corner terrain. Both maps are 40 grids ∗ 40 grids in size, as depicted in Fig. 3.



(**a**) map 1 (**b**) map 2

Figure 3: The maps of 40 grids ∗ 40 grids

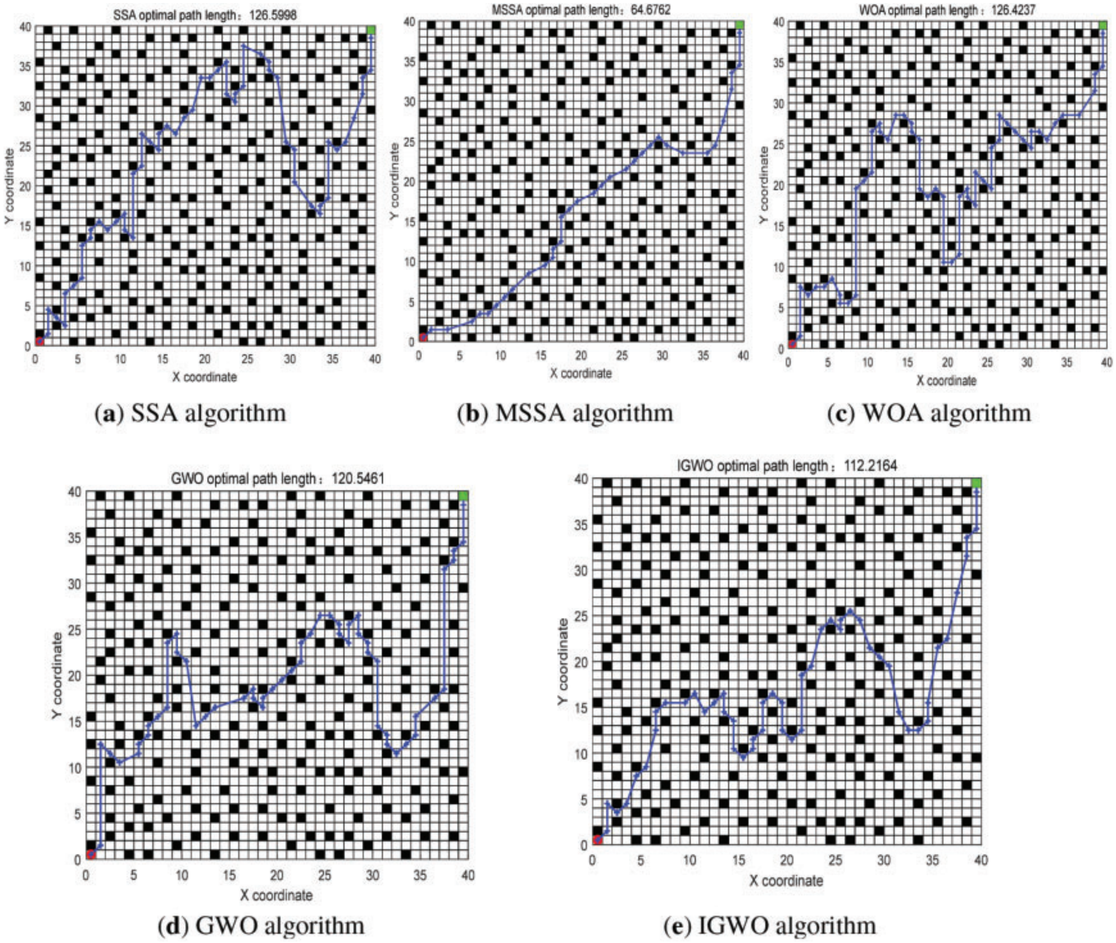
5.1.1 Optimal Path Planning for Map 1

(1) Qualitative description of the optimal path

In the case of map 1, the optimal path planning results of five comparative algorithms are shown in Fig. 3. It can be intuitively observed that the MSSA algorithm plans the optimal path more smoothly compared to other algorithms. There is no significant back and forth on the trajectory, thereby reducing the total length of the optimization path. Indeed, the optimal path planned by the MSSA algorithm is proven to be the shortest. From the simulation results, the optimal path lengths of Figs. 4a–4e are 126.5998, 64.6762, 126.4237, 120.5461, 112.2164, respectively. The MSSA algorithm provides an optimal path that is nearly half the length of the path planned by other algorithms. To more accurately evaluate the performance of these five algorithms in path planning, additional quantitative analysis is required.

(2) Qualitative analysis of the optimal path

To avoid errors caused by a single simulation result, we used the above five intelligent algorithms to plan the path for map 1 with 50 times simulation. The results, including the optimal path length, average path length, iteration times of optimal path length, and minimum inflection points, are shown in Table 5.



(**a**) SSA algorithm (**b**) MSSA algorithm (**c**) WOA algorithm

(**d**) GWO algorithm (**e**) IGWO algorithm

Figure 4: The optimal path planning for map 1

Table 5: Comparison of map 1 results

Algorithm	The optimal path length	Average path length	Iteration times of the optimal path length	Minimum inflection points
SSA	125.5	129.6	196	50
MSSA	68.7	70.1	78	25
WOA	128.5	134.8	73	48
GWO	120.1	125.4	195	48
IGWO	110.3	113.5	185	47

Table 5 displays the lengths of the optimal paths provided by the SSA algorithm, MSSA algorithm, WOA algorithm, GWO algorithm, and IGWO algorithm, which are 125.5, 68.7, 128.5, 120.1, and 110.3, respectively. The path planned by the MSSA algorithm experiences a substantial reduction in length compared to the paths planned by other algorithms, showcasing a clear advantage. Compared to the SSA algorithm, the WOA algorithm, the GWO algorithm, and the IGWO algorithm, the MSSA algorithm effectively shortens the optimal path length by 45.25%, 46.53%, 42.79%, and 37.71%. Regarding the average path length, the order of path planning length from short to long is consistent with the order of optimal path length. When compared with the SSA algorithm, the WOA algorithm, the GWO algorithm, and the IGWO algorithm, the MSSA algorithm reduces the average path length by 45.91%, 47.99%, 44.09%, and 38.23%. These results highlight the advantages of the MSSA algorithm in path planning.

With regard to the number of iterations required for the optimal paths, the order of iteration times from least to most is as follows: the WOA algorithm, the MSSA algorithm, the IGWO algorithm, the GWO algorithm, and the SSA algorithm. While the number of iterations of the MSSA algorithm is only 5 times more than that of the WOA algorithm, the average path distance planned by the MSSA is reduced by 47.99% compared to the WOA algorithm. It demonstrates that the WOA algorithm achieves fewer iterations at the expense of sacrificing path length. As for the number of inflection points in the path, the MSSA algorithm has the least number of inflection points, which is about half the number of inflection points given by other algorithms. The number of inflection points in the paths of SSA, WOA, GWO, and IGWO algorithms is in the range of 47 to 50. The path planning results visually demonstrate that these paths are not smooth. In summary, the path planning results of the MSSA algorithm in map 1 are significantly higher than other comparative algorithms.

5.1.2 Optimal Path Planning for Map 2

(1) Qualitative description of the optimal path

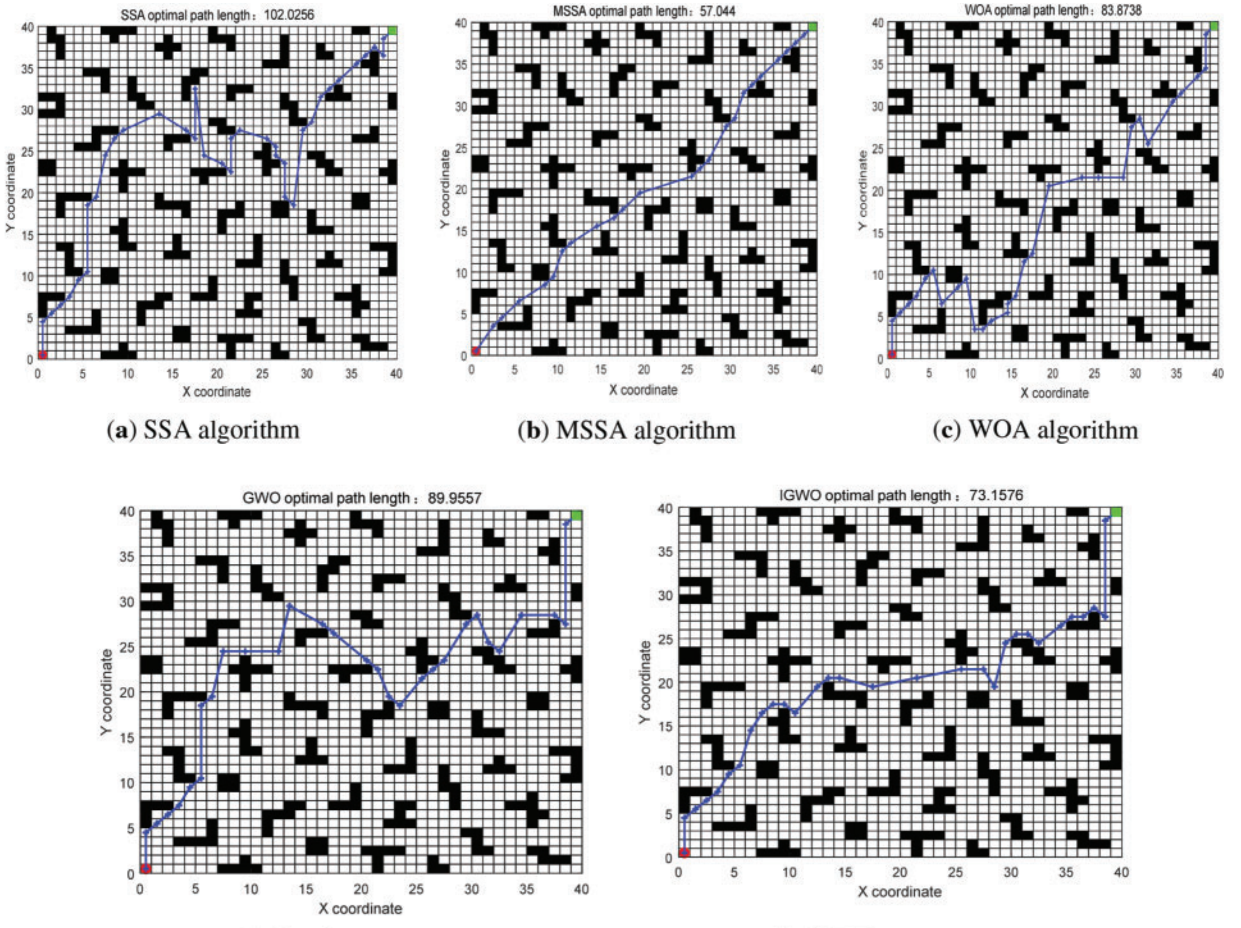
The optimal path planning results of five comparative algorithms in map 2 are shown in Fig. 4. It can be intuitively seen from Fig. 5 that compared to other algorithms, the optimal path planned by the MSSA algorithm is smoother, approaching a straight line. This conclusion is consistent with the qualitative description conclusion for map 1. Similarly, we quantitatively analyzed the performance of different algorithms for map 2 optimal path planning.

(2) Qualitative analysis of the optimal path

To avoid errors caused by a single simulation result, we used the above five intelligent algorithms to plan the path for map 2 with 50 times simulation. The results, including the optimal path length, average path length, iteration times of optimal path length, and minimum inflection points, are shown in Table 6.

In Table 6, the optimal path lengths of these five algorithms, from shortest to longest, are the MSSA algorithm, the IGWO algorithm, the GWO algorithm, the WOA algorithm, and the SSA algorithm. Compared to the SSA algorithm, the WOA algorithm and the GWO algorithm, the path planned by the MSSA algorithm is significantly shortened with 44.48%, 33.76%, and 37.65%, respectively. The IGWO algorithm performs better in map type 2 than in map type 1, and the difference in path length planned by the MSSA algorithm is reduced by approximately 21.9%.

Compared to the SSA algorithm, the WOA algorithm, the GWO algorithm, and the IGWO algorithm, the average path length planned by the MSSA algorithm is significantly shortened with 42.96%, 33.99%, 38.23%, and 22.09%, respectively. Although the MSSA algorithm's advantage is slightly diminished in map 2 compared to map 1, it still surpasses other comparative algorithms. Concerning optimal path iteration times, the order of iteration times from least to most is MSSA, WOA, GWO, SSA, and IGWO. Meantime the MSSA algorithm has the least number of inflection points and the path is the smoothest. In summary, the path planning results of the MSSA algorithm in map 2 are significantly better than other comparative algorithms. The analysis of the path planning results for map 1 and map 2 demonstrates the high application value of the MSSA algorithm.



(a) SSA algorithm **(b)** MSSA algorithm **(c)** WOA algorithm

(d) GWO algorithm **(e)** IGWO algorithm

Figure 5: Optimal path planning results for map 2

5.2 The Path Planning Results of a Multi-Exit Evacuation Scenario

There are certain differences between the map scene in Section 5.1 and the real evacuation scene. To reflect the application effect of the improved algorithm proposed in the real evacuation scene, we further built a multi-exit evacuation scenario model. Due to a comprehensive comparison of the path planning results of different algorithms in Section 5.1, only the basic sparrow search algorithm and the improved sparrow search algorithm have been compared in terms of path planning effects. In this section, we construct a simplified grid map of a teaching building, shown in Fig. 6, and apply the SSA algorithm and MSSA algorithm to plan evacuation paths for its four exits. We ultimately compare and analyze the length of evacuation paths.

Table 6: Comparison of map 2 results

Algorithm	The optimal path length	Average path length	Iteration times of the optimal path length	Minimum inflection points
SSA	101.4	104.5	168	31
MSSA	56.3	59.6	48	10
WOA	85.0	90.3	50	22
GWO	90.3	96.5	113	28
IGWO	72.1	76.5	185	21

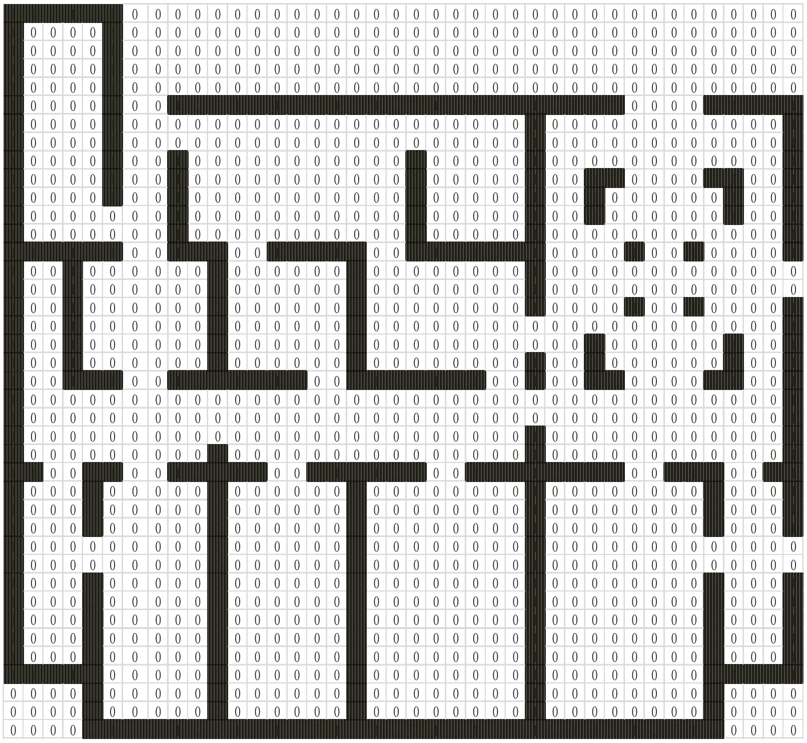
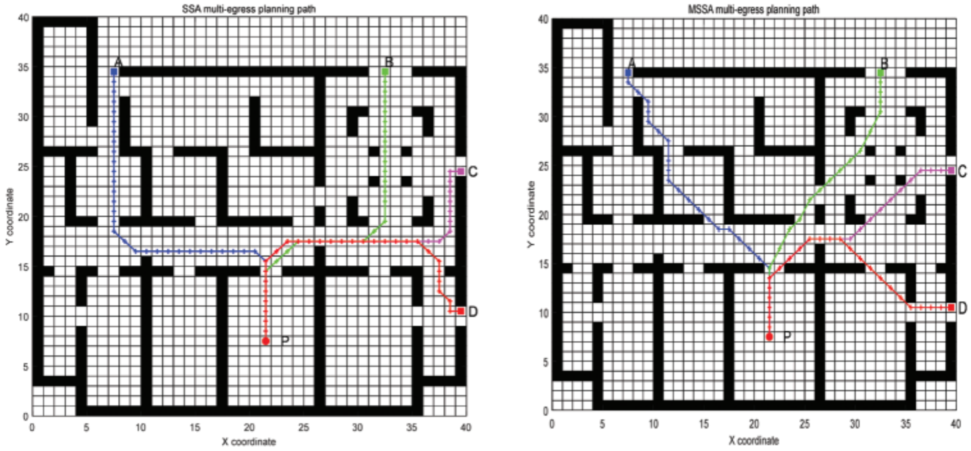


Figure 6: Multi-exit architectural model simplified grid map

The evacuation path planning results of the SSA algorithm and MSSA algorithm are shown in Fig. 7, where P is the starting position for evacuation, and A, B, C, and D are the evacuation exits.

From Fig. 7a, it is evident that in the multi-exit evacuation scene, the planned paths follow the contours of the building's inherent terrain due to the limited search ability of the SSA algorithm. Conversely, the MSSA algorithm improves its global and local search capabilities through improved strategies, allowing it to search for the shortest distance within the building's terrain for path planning. In Fig. 7b, there are some longer diagonal paths. According to the principle of the shortest diagonal between two points, the path planned in Fig. 7b appears to be shorter than the path planned in Fig. 7a.

The quantitative results of the evacuation path length planned by the SSA algorithm and MSSA algorithm are shown in Table 7. The PA path, PB path, PC path, and PD path in Table 7 are the evacuation paths from the evacuation starting position P to the four evacuation exits A, B, C, and D, respectively.


SSA multi-egress planning path
MSSA multi-egress planning path
Y coordinate
X coordinate
A
B
C
D
P
0
5
10
15
20
25
30
35
40

(**a**) SSA algorithm (**b**) MSSA algorithm

Figure 7: Path planning of different algorithms in a multi-exit teaching building

Table 7: Comparison results of the length of multi-exit evacuation path planning

Algorithm	PA path length	PB path length	PC path length	PD path length
SSA	39.2	35.1	33.2	32.1
MSSA	33.4	30.7	28.6	28.6

The path planning results of the four evacuation exits demonstrate that the MSSA algorithm has significantly shortened the path length compared to the SSA algorithm, shown in Table 7. Compared with the original SSA algorithm, the PA, PB, PC, and PD path planning results reveal that the MSSA algorithm reduces the path length by 14.8%, 12.5%, 13.9%, and 10.9%, respectively. This reaffirms the effectiveness and practical applicability of the MSSA algorithm.

6 Conclusion

Evacuation path planning has always been a concern of researchers in the security field. This paper applies the sparrow search algorithm to solve this problem. Through several improvement strategies, the search ability and convergence speed of the sparrow search algorithm are improved. To enhance the global search ability, the Golden Sine Algorithm and nonlinear weight factor were introduced in the update stage of the discoverer's position. To improve the local search ability, a mountain climbing mechanism was adopted for local search, while Gaussian-Cauchy perturbation was introduced to enhance the overall search accuracy and the capacity to jump out of the local optimal value. Then a series of standard function test sets were employed to evaluate the optimization ability of the proposed algorithm as well as the other four intelligent algorithms. The results, including the optimal value, mean value, and variance, showed that the MSSA algorithm exhibited good solving capabilities and stability compared to the other algorithms. Ultimately, the MSSA algorithm and other four comparison algorithms were applied to plan the evacuation paths on different evacuation maps using the gird method. The results include the optimal path length, average path length, iteration times of optimal path length, and minimum inflection points. Important results and conclusions are shown below:

(1) For map 1, compared with the other four comparison algorithms, the MSSA algorithm reduces the optimal path lengths by 45.25%, 46.53%, 42.79%, and 37.71%. Additionally, the MSSA algorithm reduces the average path lengths by 45.91%, 47.99%, 44.09%, and 38.23%.

(2) For map 2, compared with the other four comparison algorithms, the optimal path lengths planned by the MSSA algorithm are significantly shortened with 44.48%, 33.76%, 37.65%, and 21.9%. Moreover, the average path lengths planned by the MSSA algorithm are significantly reduced with 42.96%, 33.99%, 38.23%, and 22.09%, respectively.

(3) In the multi-exit evacuation scenario, compared with the results planned by the original SSA algorithm, the lengths of PA, PB, PC, and PD paths show that the MSSA algorithm reduces the path lengths by 14.8%, 12.5%, 13.9%, and 10.9%, separately.

These results demonstrate that the MSSA algorithm had significant advantages in evacuation path planning over the comparative algorithm, with strong potential for practical applications. However, there are still some limitations in this study. The advantage of the MSSA algorithm compared to the SSA algorithm is not obvious when solving the fixed dimension test function. Additionally, the current research primarily focuses on optimizing path length, neglecting the multi-objective optimization problem and three-dimensional spatial environment, which aligns more closely with real-world demands.

Acknowledgement: The authors wish to express their appreciation to the reviewers for their helpful suggestions which greatly improved the presentation of this paper.

Funding Statement: The present study was supported by National Natural Science Foundation of China (71904006), Henan Province Key R&D Special Project (231111322200), the Science and Technology Research Plan of Henan Province (232102320043, 232102320232, 232102320046), the Natural Science Foundation of Henan (232300420317, 232300420314).

Author Contributions: Xiaoge Wei: Conceptualization, Writing—original draft, Methodology, Funding acquisition; Yuming Zhang: Conceptualization, Data curation, Data analysis; Huaitao Song: Writing—review & editing, Funding acquisition; Hengjie Qin: Writing—review & editing; Guanjun Zhao: Writing—review & editing. All authors have read and agreed to the published version of the manuscript.

Availability of Data and Materials: The datasets used and/or analyzed during the current study are available from the corresponding author on reasonable request.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

1. Liang, B., Wal, C. N., Xie, K., Chen, Y., Brazier F, M. T. et al. (2023). Mapping the knowledge domain of soft computing applications for emergency evacuation studies: A scientometric analysis and critical review. *Safety Science, 158,* 105955.
2. Song, Q., Li, S., Yang, J., Bai, Q., Hu, J. et al. (2021). Intelligent optimization algorithm-based path planning for a mobile robot. *Computational Intelligence and Neuroscience, 2021,* 8025730. https://doi.org/10.1155/2021/8025730
3. Astri, R., Sularno, S. (2020). Implementation of A-star algorithm for searching routes near the tsunami evacuation shelter point. *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi), 4(2),* 254–259.
4. Liu, L., Wang, B., Xu, H. (2022). Research on path-planning algorithm integrating optimization A-star algorithm and artificial potential field method. *Electronics, 11(22),* 3660.
5. Guo, J., Zhang, H., Sun, G. (2021). Transportation path optimization of modern logistics distribution considering hybrid tabu search algorithm. *Advances in Multimedia, 2021,* 8375780.
6. Kumar, A., Kumar, D., Jarial, S. (2017). A review on artificial bee colony algorithms and their applications to data clustering. *Cybernetics and Information Technologies, 17(3),* 3–28.
7. Liu, X., Wei, X., Huang, H. (2021). An improved grey wolf optimization algorithm and its application in path planning. *IEEE Access, 9,* 121944–121956. https://doi.org/10.1109/ACCESS.2021.3108973
8. Wu, L., Huang, X. D., Cui, J. G., Liu, C., Xiao, W. S. (2023). Modified adaptive ant colony optimization algorithm and its application for solving path planning of mobile robot. *Expert Systems with Applications, 215,* 119410.
9. Zhang, Y., Wu, L., Wang, S. (2013). UCAV path planning by fitness-scaling adaptive chaotic particle swarm optimization. *Mathematical Problems in Engineering, 2013,* 705238. https://doi.org/10.1155/2013/705238
10. Zhang, Y. D., Wu, L. N., Wei, G., Wang, S. H. (2011). A novel algorithm for all pairs shortest path problem based on matrix multiplication and pulse coupled neural network. *Digital Signal Processing, 21 (4),* 517–521. https://doi.org/10.1016/j.dsp.2011.02.004
11. Xing, Y., Wang, X., Shen, Q. (2021). Test case prioritization based on artificial fish school algorithm. *Computer Communications, 180,* 295–302.
12. Aydilek, I. B. (2018). A hybrid firefly and particle swarm optimization algorithm for computationally expensive numerical problems. *Applied Soft Computing, 66,* 232–249.
13. Colorni, A., Dorigo, M., Maniezzo, V. (1991). Distributed optimization by ant colonies. *Proceedings of the First European Conference on Artificial Life*, pp. 134–142. Paris, France.
14. Pan, W. T. (2012). A new fruit fly optimization algorithm: Taking the financial distress model as an example. *Knowledge-Based Systems, 26,* 69–74.
15. Mirjalili, S., Mirjalili, S. M., Lewis, A. (2014). Grey wolf optimizer. *Advances in Engineering Software, 69,* 46–61.
16. Mirjalili, S., Lewis, A. (2016). The whale optimization algorithm. *Advances in Engineering Software, 95,* 51–67.
17. Xue, J., Shen, B. (2020). A novel swarm intelligence optimization approach: Sparrow search algorithm. *Systems Science & Control Engineering, 8(1),* 22–34.
18. Song, J., Jin, L., Xie, Y., Wei, C. (2021). Optimized XGBoost based sparrow search algorithm for short-term load forecasting. *2021 IEEE International Conference on Computer Science, Artificial Intelligence and Electronic Engineering (CSAIEE)*, pp. 213–217. South Carolina, USA, IEEE.
19. Wang, X., Liu, J., Hou, T., Pan, C. (2022). The SSA-BP-based potential threat prediction for aerial target considering commander emotion. *Defence Technology, 18(11),* 2097–2106.
20. Liu, T., Yuan, Z., Wu, L., Badami, B. (2021). Optimal brain tumor diagnosis based on deep learning and balanced sparrow search algorithm. *International Journal of Imaging Systems and Technology, 31(4),* 1921–1935.

21. Peng, Y., Liu, Y., Li, Q. (2020). The application of improved sparrow search algorithm in sensor networks coverage optimization of bridge monitoring. *MLIS,* 416–423.

22. Lei, Y., De, G., Fei, L. (2020). Improved sparrow search algorithm based DV-Hop localization in WSN. *2020 Chinese Automation Congress (CAC)*, pp. 2240–2244. Shanghai, China, IEEE

23. Zhang, T., Chen, J., Li, F., Zhang, K., Lv, H. et al. (2022). Intelligent fault diagnosis of machines with small & imbalanced data: A state-of-the-art review and possible extensions. *ISA Transactions, 119,* 152–171.

24. Zhang, Z., He, R., Yang, K. (2022). A bioinspired path planning approach for mobile robots based on improved sparrow search algorithm. *Advances in Manufacturing, 10(1),* 114–130.

25. Jiang, Z., Ge, J., Xu, Q., Yang, T. (2021). Fast trajectory optimization for gliding reentry vehicle based on improved sparrow search algorithm. *Journal of Physics: Conference Series, 1986(1),* 012114.

26. Liu, G., Shu, C., Liang, Z., Peng, B., Cheng, L. (2021). A modified sparrow search algorithm with application in 3d route planning for UAV. *Sensors, 21(4),* 1224.

27. Raivi, A. M., Huda, S. A., Alam, M. M., Moh, S. (2023). Drone routing for drone-based delivery systems: A review of trajectory planning, charging, and security. *Sensors, 23(3),* 1463.

28. Huang, Z., Chen, C., Pan, M. (2020). Multiobjective UAV path planning for emergency information collection and transmission. *IEEE Internet of Things Journal, 7(8),* 6993–7009.

29. Liang, J., Wang, H. (2022). Study on building fire evacuation path planning based on improved ant colony algorithm. *Journal of System Simulation, 34(5),* 1044–1053.

30. Cao, Y., Luo, C., Liu, Y., Teng, S., Xin, G. (2021). Path intelligent optimization for dense crowd emergency evacuation in heritage buildings. *Journal of Cultural Heritage, 47,* 180–187.

31. Yue, Y., Cao, L., Lu, D., Hu, Z., Xu, M. et al. (2023). Review and empirical analysis of sparrow search algorithm. *Artificial Intelligence Review, 56(10),* 10867–10919.

32. Xin, L. Y. U., Xiaodong, M., Jun, Z. A. H. N. G., Zhen, W. (2021). Chaos sparrow search optimization algorithm. *Journal of Beijing University of Aeronautics and Astronautics, 47(8),* 1712–1720.

33. Wei, X., Zhang, Y., Zhao, Y. (2023). Evacuation path planning based on the hybrid improved sparrow search optimization algorithm. *Fire, 6,* 380.

34. Tanyildizi, E., Demir, G. (2017). Golden sine algorithm: A novel math-inspired algorithm. *Advances in Electrical and Computer Engineering, 17(2),* 71–78.

35. Liu, C., Wu, L., Huang, X., Xiao, W. (2022). Improved dynamic adaptive ant colony optimization algorithm to solve pipe routing design. *Knowledge-Based Systems, 237,* 107846.