frontiers | Frontiers in Environmental Science

TYPE Methods
PUBLISHED 28 November 2022
DOI 10.3389/fenvs.2022.1055807

OPEN ACCESS

EDITED BY
Jiaxing Cui,
Central China Normal University, China

REVIEWED BY
Jiqiang Niu,
Xinyang Normal University, China
Xiaojian Wei,
East China University of Technology, China

*CORRESPONDENCE
Qing He,
qhe@gzu.edu.cn
Liu Yang,
lyang3@gzu.edu.cn

SPECIALTY SECTION
This article was submitted to Environmental Informatics and Remote Sensing,
a section of the journal
Frontiers in Environmental Science

RECEIVED 28 September 2022
ACCEPTED 01 November 2022
PUBLISHED 28 November 2022

CITATION
Yang Y, He Q and Yang L (2022), UAV trajectory planning based on an improved sparrow optimization algorithm with multi-strategy integration.
Front. Environ. Sci. 10:1055807.
doi: 10.3389/fenvs.2022.1055807

UAV trajectory planning based on an improved sparrow optimization algorithm with multi-strategy integration

Yu Yang[1], Qing He[1]* and Liu Yang[2]*

[1]College of Big Data and Information Engineering, Guizhou University, Guiyang, China, [2]School of Public Administration, Guizhou University, Guiyang, China

Real-time monitoring of urban high-altitude data is an important goal in the construction and development of smart cities today. However, with the development of modern cities, the monitoring space becomes complicated and narrow because of the different building heights and no-fly zones, which makes UAV trajectory planning more difficult. In this paper, a multi-strategy sparrow search algorithm (MSSA) is proposed to solve the UAV trajectory planning problem in a three-dimensional environment. The algorithm aims to minimize the flight distance and maximize the use efficiency of the UAV. First, the improved algorithm employed a reverse-learning strategy based on the law of refraction to improve the search range and enhance the optimization performance. Second, we introduced a random step size generated by Levy flight into the position update strategy of the participant. The algorithm accuracy and speed of convergence were improved by the randomness feature. Finally, the algorithm incorporated the Cauchy mutation to improve the scout position, which enhanced its ability to jump out of the local optimum of the algorithm. Sixteen benchmark test functions, Wilcoxon rank sum test, and 30 CEC2014 test function optimization results demonstrated that MSSA had better optimization accuracy, convergence speed, and robustness than the comparison algorithms. In addition, the proposed algorithm was applied to the UAV trajectory planning problem in different complex 3D environments. The results confirmed that the MSSA outperformed the other algorithms in complex 3D trajectory planning problems.

KEYWORDS

smart cities, trajectory planning, improved sparrow search algorithm, UAV, three-dimensional

Introduction

With the continuous development of information technology and smart cities (Van Steen and Leiba, 2018), UAV technology plays an important role in urban emergencies and transportation networks. Due to their usefulness, reliability, safety, and relatively low cost (Rodríguez et al., 2021), UAVs (commonly referred to as drones) have become an indispensable part of the operations of a smart city. In research on UAV technology, trajectory planning is the key to the whole system. The main technical difficulty is to find a feasible way to avoid obstacles when the starting and ending points are known (Zhao et al., 2018). At present, many cities around the world are facing the problem of traffic congestion. In particular, there are periodic or sudden increases in the number of vehicles during peak hours, large events, construction work or accidents. In an emergency, the use of drones can assist traffic police in implementing security in smart cities, and rapid deployment can collect real-time information (Qadir, 2021). However, due to the complexity of the flight environment in modern urban spaces, UAV monitoring tasks are becoming more and more difficult (Mohamed, 2020). Therefore, how to outfit a UAV so that it can find a flight path that avoids obstacles but still maintains the shortest distance to the destination is a main focus of today's UAV research.

The process of UAV trajectory planning is a combinatorial optimization problem. Currently, UAVs have a low efficiency in completing tasks in complex areas because of the large scale of the mathematical model and its high computational complexity. The complexity and difficulty of real-world optimization problems continue to increase, as they are subject to strong constraints and require long computation times, non-convexity, and wide search space (Shin and Bang, 2020). Path length is the primary factor to be considered in engineering scenarios, and optimizing path is of great research significance for improving flight efficiency. Path optimization aims to maximize the execution efficiency of the UAV within permissible limits. The methods for deducing the optimal power flow in trajectory planning can be divided into traditional optimization algorithms and metaheuristic algorithms. Traditional optimization algorithms mainly include the gradient descent method Salgado et al., 1990), the Newton method (Tinney ea al., 1967), linear programming (Olofsson et al., 1995), and the interior point method (Momoh, 1999). These algorithms are characterized by their use of the objective function to solve the first- or second-order gradient of control variables. Traditional optimization algorithms are generally trapped in local optima; hence, the optimization results depend greatly on the initial value in solving large-scale problems. In recent years, researchers have proposed many metaheuristic algorithms by simulating various biological behaviors and physical phenomena in nature. The metaheuristic algorithms commonly include Archimedes optimization algorithm (AOA) (Hashim ea al., 2021), the tunicate swarm algorithm (TSA) (Kaur et al., 2020), Aquila optimization (AO) (Abualigah et al., 2021), Harris hawks optimization(HHO) (Heidari et al., 2019), ant colony optimization (ACO) (Zhang et al., 2015), grey wolf optimizer (GWO) (Mirjalili et al., 2014), differential evolution (DE) (Price, 2013), and particle swarm optimization (PSO) (Marini and Walczak, 2015).

Metaheuristic algorithms are widely used to solve problems related to power system optimization because of their simple structure, few adjustment parameters, and lack of need for gradient information. Wen et al. (2022) proposed a novel heuristic algorithm based on a three-dimensional (3D) UAV deployment scheme that could be used by a number of covered users without increasing the number of UAVs. Fan et al. (2022b) proposed an improved RRT algorithm based on the process of extending the random tree, and introduced ACO to make the planning path asymptotically optimal. Jia et al. (2022) described a UAV path coverage algorithm based on a 'greedy' strategy and ACO (Zhang et al., 2015) to minimize flight time and maximize coverage. Shin and Bang (2022) offered an improved PSO algorithm for path optimization. Belge et al. (2022) developed a new UAV trajectory planning algorithm for optimal path planning and tracking using HHO and GWO. Zhang et al. (2021) proposed an adaptive convergence factor adjustment strategy and an adaptive weight factor to update the individual's position based on GWO. Zhang et al. (2015) created an improved constrained DE algorithm to generate an optimal feasible route. Chang et al. (2021) introduced Q-learning to improve the dynamic window algorithm and increase its success rate for path planning in an unknown mountainous environment. However, the calculations necessary to use this algorithm are more complex, and beyond the low computational power of the UAV; the local

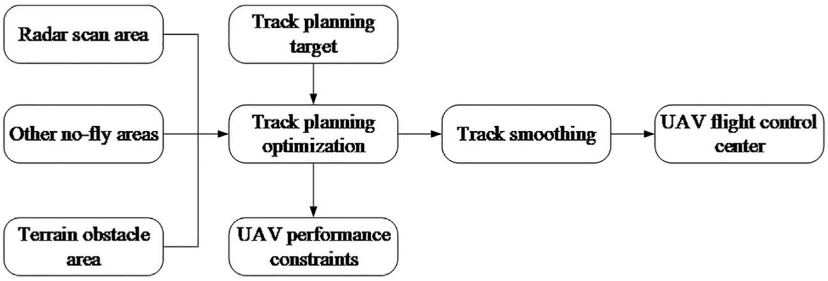

FIGURE 1
Flow chart of the trajectory planning system.

path planning strategy is not applicable to global optimization.

The sparrow search algorithm (SSA), a swarm intelligence optimization algorithm proposed by Xue and Shen (2020), combines simplicity with flexibility. It is very effective for solving highly nonlinear, multi-variate, and multimodal function optimization problems. Many researchers have studied the SSA algorithm and confirmed that it outperforms the GWO, PSO (Poli et al., 2007), and GA (Whitley, 1994) in solving numerous types of optimization problems. The SSA has been used in many other fields. Wu et al. (2021) reported the application of GGSC-SSA to solve the traveling salesman problem (TSP); Fan et al. (2022a) even utilized SSA to improve the quality of medical images; but few published reports exist on the application of SSA to UAV trajectory planning. Here we propose a new metaheuristic optimization algorithm based on the SSA and referred to as MSSA. Since the B-spline curve cannot guarantee absolute accuracy of the interpolation points (Thompson and Patel, 1987), we used the cubic spline interpolation method to smooth the path. First, we established a 3D environmental model for UAV trajectory planning that included reference terrain, obstacle areas, and threat areas. Second, a comprehensive cost evaluation model of UAV flight was proposed as the objective function. The path was smoothed by the method of cubic spline interpolation to obtain an optimal trajectory. Lastly, we analyzed the results and verified the effectiveness and feasibility of the proposed algorithm in planning the UAV's trajectory in the 3D model.

Study area and environment modelling

Overview of the study area

With the leapfrog development of information science, various new technologies employing AI for advanced communication and control have been put forward, many of which have been applied in UAV operating systems, providing a foundation for the rapid deployment of UAVs for numerous uses. At present, UAVs have become an important accessory for air power in the military, which can perform battlefield reconnaissance, supply delivery missions, and enemy target strikes. In addition to military applications, civilian applications of UAVs have also been accelerating in a number of areas, including traffic supervision, disaster relief, inspections, and scientific data gathering. The UAV trajectory problem is a multi-constraint combinatorial optimization problem. Due to the size of the mathematical model and the complexity of the calculation, a suitable cost function and an effective trajectory planning method are both crucial to the efficient implementation of UAVs.

Description of trajectory planning

Finding an optimal path using planning algorithms is the main goal of UAV trajectory planning, and this path must meet performance indicators and overcome limitations. The UAV may encounter several hurdles throughout this trajectory planning process, including terrain threats, fire, no-fly zones, and performance limitations imposed by the equipment itself (Bagherian and Alos, 2015). For testing of this algorithm, we held the UAV's speed constant and kept track of the distribution of the mission environment's peaks and no-fly zones to make the computation model simpler. The challenge of trajectory planning was changed into a static space routing mission problem. The trajectory planning system is depicted in Figure 1. Figure 1 illustrates the components of the UAV trajectory planning process, which included environmental modelling, cost function definition, track optimization, and track smoothing. Our goal in this study was to establish an optimal flight path before the UAV flight mission. By sending the algorithm's best path data to the UAV's flight master control system, the flight trajectory planning will be accomplished successfully.

Task environment modelling

According to the 3D space environment there were significant changes in altitude and terrain complexity along the route (Dübel and Schumann, 2017), and the flight path may be categorized into areas of plains, mountainous regions, and hilly areas. A large number of complex factors need to be considered if the flight must pass through a mountainous or hilly area. The UAV needs to adjust the travel direction and the flight altitude continuously in those complex environments, and trajectory planning must be carried out in 3D space. In this study, we developed two distinct settings for UAV trajectory planning and used a function simulation approach to characterize landform properties. Eq. 1 displays the function expression:

$$z(x, y) = \sin(y + a) + b\sin(x) + c\cos\left(d\sqrt{y^2 + x^2}\right) + e\cos(y) + f\sin\left(f\sqrt{y^2 + x^2}\right) + g\cos(y) \tag{1}$$

where (x, y) is the point coordinate of the terrain projected onto the horizontal plane and z is the height of the corresponding point coordinate on the Z axis. In Eq. 1, a, b, c, d, e, f, g, and h are constant coefficients, and the different topography features can be obtained by changing the size of the constant coefficients in the modelling process. In this test flight, we simulated the geographical environment, such as mountains and hills, by superimposing the mountain model on the base terrain. The

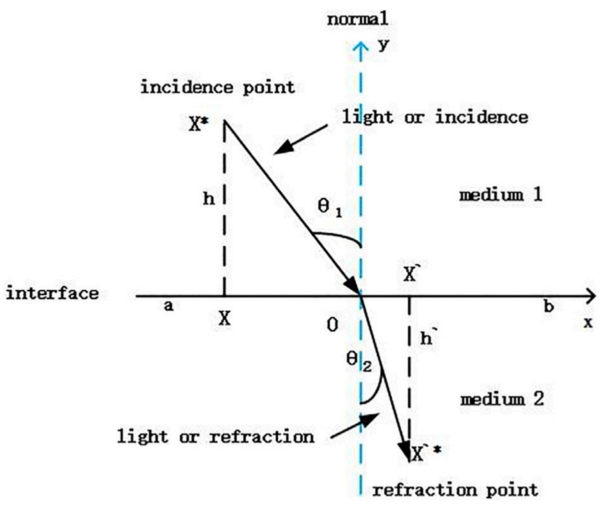

FIGURE 2
Refraction-learning process in one-dimensional space.

mathematical expression of the topography model is shown in Eq. 2:

$$h(x,y) = \sum_i h_i \exp\left[-\frac{(x_i - x_{oi})^2}{a_i^2} - \frac{(y_i - y_{oi})^2}{b_i^2}\right] + h_o \qquad (2)$$

In Eq. 2, h_o and h_i represent the height of the reference terrain and the ith peak, respectively, (x_{oi}, y_{oi}) represent the central coordinate position of the ith peak, and a_i and b_i are the slopes of the ith mountain along the X and Y axes, respectively. The peaks can show different length and width characteristics by adjusting the value of those parameters. We can obtain Eq. 3 from Eqs 1, 2:

$$Z(x,y) = \max[z(x,y), h(x,y)] \qquad (3)$$

In practical situations, UAVs frequently encounter areas with tall buildings and trees that threaten flight safety, or no-fly areas where UAVs have to avoid obstacles. Therefore, we included a certain number of threat areas in the UAV trajectory plan to judge the obstacle avoidance performance of the algorithm. Graphically, we represented a danger area as a cylinder with a radius of R to simplify the model. The center position of each cylinder affords the greatest threat to the UAV, and the threat decreases from the center to the outside.

UAV track planning modeling

When completing challenging jobs, the UAV trajectory planner must take into consideration the inherent performance restrictions of the drone based on environmental modeling. The final results produced by the algorithm can be made to comply with the specifications and guarantee that the intended flight route is valid with a suitable design of the trajectory evaluation function. We devised a sophisticated track evaluation algorithm to plan the UAV trajectory based on real circumstances. The indicators that most affected the performance of the UAV included track length, flight height, minimum step size, corner cost, and maximum climb angle.

Trajectory planning is inseparable from searching for the shortest path because the length of the track is very important for successful trajectory planning. Obviously, the shortest route can save on fuel and time and reduce the chance occurrence of unknown threats. We defined the path as the value of the distance from the starting to the ending point. Suppose a complete route has n nodes, the distance between the ith and the $i+1$-th waypoint is expressed as l_i, the coordinates of these two flight points are expressed as (x_i, y_i, z_i) and $(x_{i+1}, y_{i+1}, z_{i+1})$, denoted the two points as $g(i)$ and $g(i+1)$, respectively. The trajectory needs to satisfy the following conditions in Eq. 4:

$$\begin{cases} li = \left\|g(i+1) - g(i)\right\|_2 \\ L_{\mathrm{path}} = \sum_{i=1}^{n-1} l_i \end{cases} \qquad (4)$$

The UAV will run the risk of crashing or being shot down if it is unable to avoid obstacles or flying into a hazard region, which is indicated by the L_{path} being $L_{path} = \infty$. Because infinite functions are challenging to depict in real-world situations, we deal with them in a penalizing approach. The UAV should fly as low as it can in the real world to avoid potential radar detection. However, it is crucial to choose a steady flight altitude because a low flight altitude would increase the probability of the UAV colliding with trees, mountains, or the ground. The flying altitude should not change much because a constant altitude eases the strain on the control system and conserves fuel. To make the UAV flight safer, the flight height model given in Eq. 5:

$$\begin{cases} h_{\mathrm{height}} = \sqrt{\frac{1}{n}\sum_{i=0}^{n-1} (z(i) - \bar{z})^2} \\ \bar{z} = \frac{1}{n}\sum_{i=0}^{n-1} z(i) \end{cases} \qquad (5)$$

The maneuverability of a UAV is limited by its corner cost function. During the flight of the UAV trajectory planning, the turning angle should not be greater than the preset maximum turning angle, because of the turning angle size will affect the flight stability. In this paper, we set the maximum turning angle to φ and the current turning angle to θ, and a_i is the vector of the ith route segment. The corner formula is shown in Eq. 6:

$$\begin{cases} \cos\theta = \dfrac{a_i^T a_{i+1}}{|a_i||a_{i+1}|} \\ J_{turn} = \sum_{i=1}^{n} (\cos(\Phi - \cos\theta)) \end{cases} \qquad (6)$$

In Eq. 6, $|a|$ represents the length of the vector a. Through the description of the above three aspects, we established the cost function of UAV trajectory planning as follows in Eq. 7:

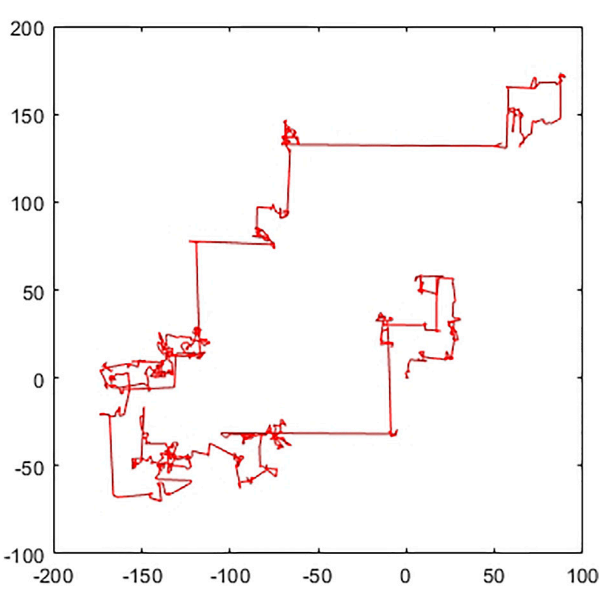

FIGURE 3
Step size of 100 Levy flights.

$$J_{cost} = w_1 L_{path} + w_2 h_{height} + w_3 J_{turn} \tag{7}$$

J_{cost} is the total cost function, and in the parameters w_i, $i = 1, 2, 3$ represent the weight of each cost function and satisfy the conditions of Eq. 8:

$$\begin{cases} w_i \geq 0 \\ \sum_{i=1}^{3} w_i = 1 \end{cases} \tag{8}$$

We obtained a track consisting of line segments by processing the total cost function effectively. However, the resulting track is only theoretically feasible, it is necessary to smooth the track to meet the actual situation. In this paper, cubic spline interpolation is used to smooth the UAV trajectory because the B-spline curve cannot guarantee the absolute accuracy of the interpolation points.

Sparrow search algorithm

The sparrow search algorithm is a new type of swarm intelligence optimization algorithm inspired by the feeding behavior of sparrows in nature. During the food search process, the sparrow population is divided into two roles: discoverer and follower. Discoverers generally make up 10%–20% of the population and lead the other individuals in the search for food. The discoverers have a high fitness and ability to expand the search range, while the remaining sparrows follow the discoverers to the destination. The population requires a particular number of sparrows to work as scouts and issue warnings to remind the populace that they can take action in time when the adversaries attack to escape the threat of natural enemies. The location update formula for the discoverer is as follows in Eq. 9:

$$x_{id}^{t+1} = \begin{cases} x_{i,j}^{t} \cdot \exp\left(\frac{-h}{\alpha \cdot T}\right) & if\ R_2 < ST \\ x_{i,j}^{t} + Q \cdot L & if\ R_2 > ST \end{cases} \tag{9}$$

In Eq. 9, h represents the current number of iterations and T is the maximum number of iterations. The value of $x_{\text{i,j}}$ denotes the current position of the ith sparrow in the jth dimension. The term, α, is a random number between 0 and 1, and Q is a random value obeying a standard normal distribution. L represents a 1×D matrix with all elements 1, alarm value $R_2 \in [0,1]$, and safety value $ST \in [0.5,1]$.

When $R_2 < ST$, it indicates that the surrounding environment is currently in a safe state, and the discoverers can search for food on a wide scale. If $R_2 > ST$, it means that there may be natural enemies in the surrounding environment, and the discoverers will quickly lead the population from the current position to avoid predators. Followers update their position according to their fitness ranking, and the positional update is described in Eq. 10:

$$x_{i,j}^{T+1} = \begin{cases} Q \cdot \exp\left(-\frac{x_{worst}^{T} - x_{i,j}^{T}}{i^2}\right) & if\ i > \frac{n}{2} \\ x_p^{T+1} + \left|x_{x,j}^{T} - x_p^{T+1}\right| \cdot A^{+} \cdot\ L\ otherwise \end{cases} \tag{10}$$

In Eq. 10, $xt\ w$ and $x_{t+1}\ p$ represent the global worst position of the population at the tth iteration and the global optimal position of the population at the $t+1$-th iteration, respectively. A is a matrix of $1 \times D$ in which an element is only -1 or +1, with $A^+ = A^T(AA^T)^{-1}$. When $i > n/2$, it indicates that the ith participant is in a hungry state with poor fitness. In order to obtain higher energy, the participant needed to fly farther to find food. If $i < n/2$, the ith follower will find a random location near the current optimal position x_{p} to forage. The location of the scouter has been updated as shown in Eq. 11:

$$x_{i,j}^{T+1} = \begin{cases} x_{i,j}^{T} + Q \cdot \left|x_{i,j}^{T} - x_{best}^{T}\right| & f_i \neq f_g \\ x_{i,j}^{T} + K \frac{x_{i,j}^{T} - x_{worst}^{T}}{(f_i - f_w) + \varepsilon} & f_i = f_g \end{cases} \tag{11}$$

where K is a step coefficient, which is a random number in [-1, 1], and K represents the moving direction of the sparrows. Q is a value close to infinitesimal, which exists to avoid the denominator being zero. f_i represents the fitness values of the ith sparrows, f_g and f_w are the global optimal fitness and global worst fitness values within the current search scope, respectively, ε is the smallest real number, preventing the occurrence of 0 in the denominator. Individual sparrows face danger at the edge and approach the globally optimal sparrow

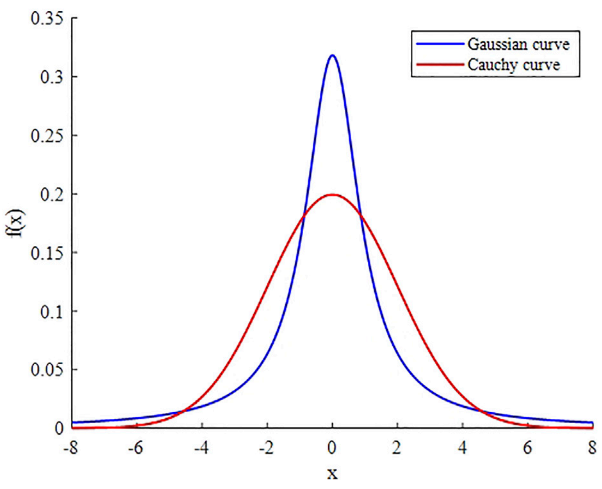

FIGURE 4
Gaussian curve and Cauchy curve distribution diagram.

when $f_i \neq f_g$. The individual sparrows in the center of the group can randomly walk among other individuals when $f_i = f_g$. This setup aims to avoid too many individuals reaching a local optimum, where the algorithm stops iterating and falls into a local optimum.

Multi-strategy sparrow search algorithm

The original SSA algorithm used a simple random function to generate the initial population, which cannot guarantee the diversity of the population and the stability of the algorithm. In the sparrow population, the foraging ability of the discoverers determines the foraging direction and foraging area of the population and also indicates the quality of the solution found by the algorithm. The formula for updating the discoverers' positions is one of the key formulas in the SSA. In the later iterations of the SSA, the sparrow population gradually approaches to the optimal individual, which leads to a lack of population diversity and a tendency toward premature convergence of the algorithm. In the SSA, it can be seen from Eq. 9 that each dimension of the individual discoverer decreases when $R2<ST$. This leads to a decrease in the population diversity of the algorithm in its later iterations and a lack of convergence accuracy. In the iterative process of the algorithm, the location update of sparrows mainly depends on the information exchanged among individuals, which easily produces population aggregation and leads to a lack of diversity. Then, according to Eq. 10, a large number of followers will flood into the search area around the finder. When they perceive that the finder has searched for a better food location, the high population density around the finder educes the diversity of population positions and easily falls into a local optimum. In summary, we propose three corresponding improvement strategies for the original SSA algorithm, which is prone to local optimal values and insufficient convergence accuracy. The specific strategies are discussed below.

Reverse learning strategy based on the law of refraction

For the SSA in the finder stage, with the continuous expansion of the search range, a broad and flexible search mechanism is the key to guiding the entire sparrow population to find food and avoid danger. When $R2 < ST$, the discoverer individuals of each dimension decrease. To better realize the leading role of the discoverer, this paper proposes a reverse learning strategy based on the law of refraction. We calculate the reverse solution of the candidate solution and select the better solution to continue the iterative calculation to effectively enhance the diversity of the algorithm and help the algorithm jump out of the local extreme value space.

As shown in Figure 2, O is the center point of $[a, b]$, $x\in[a, b]$; the global optimal individual X takes O as the center point to find its corresponding reverse individual X', where a represents the upper bound and b represents the lower bound. It can be derived from the law of refraction shown in Eq. 12:

$$\gamma = \frac{\sin\theta_1}{\sin\theta_2} = \frac{((a+b)/2 - x)/h}{(x' - (a+b)/2)/h'} \tag{12}$$

Assuming the scaling factor $\eta = h/h'$, the mathematical model of the reverse individual X' can be obtained by deriving Eq. 12 as shown in Eq. 13:

$$x' = \frac{(a+b)}{2} + \frac{(a+b)}{(2\eta\gamma)} - \frac{x}{\eta\gamma} \tag{13}$$

Extend Eq. 13 to n-dimensional space to obtain:

$$x_j^{'} = \frac{\left(a_j + b_j\right)}{2} + \frac{\left(a_j + b_j\right)}{(2\eta\gamma)} - \frac{x_j}{\eta\gamma} \tag{14}$$

In this formula, a_j and b_j represent the jth dimensional vector of the upper bound and the jth dimensional vector of the lower bound, respectively; x_j and $x' j$ represent the jth dimensional vector of X and $x^{'}$, respectively. Refraction reverse learning is performed on the optimal solution in the population, and each dimension value is mapped to the solution space to obtain a reverse solution, which not only avoids the interference between various dimensions but also expands the search range of the algorithm.

TABLE 1 Introduction to benchmark functions.

Fun no.	Name	Function type	Range	Dim	Optimal value
f_1	Sphere	Single-modal	[-100,100]	30/200/500	0
f_2	Schwefel's problem 2.22	Single-modal	[-10,10]	30/200/500	0
f_3	Schwefel's problem 1.2	Single-modal	[-100,100]	30/200/500	0
f_4	Schwefel's problem 2.21	Single-modal	[-100,100]	30/200/500	0
f_5	sum spare	Single-modal	[-10,10]	30/200/500	0
f_6	Zakharov	Single-modal	[-5,10]	30/200/500	0
f_7	Generalized Schwefel's problem 2.26	Multi-modal	[-500,500]	30/200/500	-12569.5
f_8	Generalized Rastrigin's function	Multi-modal	[-5.12,5.12]	30/200/500	0
f_9	Ackley's Function	Multi-modal	[-32,32]	30/200/500	0
f_{10}	Ceneralized Criewank function	Multi-modal	[-600,600]	30/200/500	0
f_{11}	Apline	Multi-modal	[-10,10]	30/200/500	0
f_{12}	Ceneralized penalized function 2	Fixed multi-modal	[-50,50]	30/200/500	0
f_{13}	Shekell's foxholes function	Fixed multi-modal	[-65,65]	30/200/500	1
f_{14}	Hatman's function 1	Fixed multi-modal	[0,1]	30/200/500	-3.86
f_{15}	Hatman's function 2	Fixed multi-modal	[0,1]	30/200/500	-3.32
f_{16}	Eggcrate	Fixed multi-modal	[-2π,2π]	30/200/500	0

Follower position update strategy based on Levy flight

Although the reverse learning strategy of the refraction law can help the algorithm to jump out of the local optimal value and improve the solution accuracy of the algorithm, it cannot expand the search range of the original population in the optimization process. Therefore, we introduced the Levy flight strategy to expand the follower's optimal range.

According to Eq. 10, a large number of followers in SSA will flood into the search area around the discoverers when they perceive that the finder has searched for a better food location, which has obvious convergence and makes the population density of the search area too high around the discoverers. This situation will reduce the diversity of population positions and easily fall into a local optimum. In this paper, the random step size *s* generated by Levy's flight is introduced into the follower's location update strategy, and the uncertainty of Levy's flight direction and step size is used to enhance the multiplicity of follower's search direction, thus improving the diversity of population locations and avoiding the search from falling into local optimal value. The random step size *s* of the Levy flight can be calculated by Eq. 15, and the result is shown in Figure 3:

$$\begin{cases} s = \mu \big/ |v|^{1/\beta}, \beta = 3\big/2 \\ \mu \sim N\left(0, \sigma_\mu^2\right), v \sim N\left(0, \sigma_v^2\right) \\ \sigma_\mu = \left\{ \dfrac{\Phi(1+\beta)\sin(\pi\beta/2)}{\beta \cdot \Phi[(1+\beta)/2] \cdot 2^{(\beta-1)/2}} \right\}^{1/\beta}, \sigma_v = 1 \end{cases} \tag{15}$$

Figure 3 shows that Levy flight shuttles each other due to short and long distance searches, producing random steps *s* with no definite direction or size. By introducing the Levy flight strategy, followers can both roughly search over a large range and finely search over a small range as they approach the discoverer, which can effectively avoid convergence and enhance the diversity of population locations. The formula for updating the position of the follower was changed from Eqs 10–16 with the addition of the Levy flight strategy:

$$x_{i,j}^{t+1} = \begin{cases} Q \cdot \exp\left(\dfrac{x_w^t - x_{i,j}^t}{i^2}\right) & if\ i > \dfrac{n}{2} \\ x_p^{t+1} + s \cdot \left|x_{x,j}^t - x_p^{t+1}\right| A^+ \cdot L & otherwise \end{cases} \tag{16}$$

According to Formula 10, it can be seen that the individual with lower fitness value as a follower will reset the solution to a number near 1 after updating according to the formula, which is more effective for some objective functions whose optimal convergence solution is near 1 or 0. However, in practical engineering applications, the sparrows generally fly to places with lower fitness values on the whole. At the same time, the sparrow individuals with moderate fitness values are directly replaced by the current best individual. Although the convergence speed is improved to a certain extent, it will waste the search area of this part of the sparrow population and reduce the search accuracy. Comprehensively comparing the characteristics of the individual fitness values of followers, sparrows with moderate fitness values search in the direction of the best sparrow individual according to the current search area, while sparrow individuals with low fitness value fly to search

TABLE 2 Comparison of optimization results of different improvement strategies for SSA.

Fun no.	Algorithm	Optimal	Worst	Mean	Standard
f_1	SSA	0.00E+00	1.03E-50	3.44E-52	1.89E-51
	SSA1	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	SSA2	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	SSA3	0.00E+00	1.32E-65	4.45E-67	2.41E-66
	MSSA	0.00E+00	0.00E+00	0.00E+00	0.00E+00
f_2	SSA	0.00E+00	2.28E-21	7.59E-23	4.16E-22
	SSA1	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	SSA2	0.00E+00	5.82E-272	3.09E-273	0.00E+00
	SSA3	0.00E+00	5.56E-30	1.89E-31	1.01E-30
	MSSA	0.00E+00	0.00E+00	0.00E+00	0.00E+00
f_3	SSA	0.00E+00	4.72E-63	1.61E-64	8.62E-64
	SSA1	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	SSA2	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	SSA3	0.00E+00	2.48E-50	8.28E-52	4.53E-51
	MSSA	0.00E+00	0.00E+00	0.00E+00	0.00E+00
f_4	SSA	0.00E+00	1.09E-23	3.65E-25	2.00E-24
	SSA1	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	SSA2	7.88E-306	1.30E-258	4.35E-260	0.00E+00
	SSA3	0.00E+00	2.20E-25	8.76E-27	4.06E-26
	MSSA	0.00E+00	0.00E+00	0.00E+00	0.00E+00
f_5	SSA	0.00E+00	1.47E-37	4.89E-39	2.68E-38
	SSA1	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	SSA2	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	SSA3	0.00E+00	6.54E-44	2.18E-45	1.19E-44
	MSSA	0.00E+00	0.00E+00	0.00E+00	0.00E+00
f_6	SSA	0.00E+00	5.29E-37	2.25E-38	9.93E-38
	SSA1	0.00E+00	5.65E-35	1.88E-36	1.03E-35
	SSA2	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	SSA3	0.00E+00	2.79E-46	9.29E-48	5.09E-47
	MSSA	0.00E+00	0.00E+00	0.00E+00	0.00E+00
f_7	SSA	-1.26E+04	-4.77E+03	-1.14E+04	1.92E+03
	SSA1	-1.26E+04	-4.13E+03	-1.02E+04	2.49E+03
	SSA2	-1.26E+04	-7.95E+03	-1.20E+04	9.45E+02
	SSA3	-1.13E+04	-3.96E+03	-9.30E+03	1.75E+03
	MSSA	-1.26E+04	-1.26E+04	-1.26E+04	2.03E-01
f_8	SSA	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	SSA1	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	SSA2	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	SSA3	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
F_9	SSA	8.88E-16	8.88E-16	8.88E-16	0.00E+00
	SSA1	8.88E-16	8.88E-16	8.88E-16	0.00E+00
	SSA2	8.88E-16	8.88E-16	8.88E-16	0.00E+00
	SSA3	8.88E-16	8.88E-16	8.88E-16	0.00E+00
	MSSA	8.88E-16	8.88E-16	8.88E-16	0.00E+00
f_{10}	SSA	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	SSA1	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	SSA2	0.00E+00	0.00E+00	0.00E+00	0.00E+00

(Continued on following page)

TABLE 2 (*Continued*) Comparison of optimization results of different improvement strategies for SSA.

Fun no.	Algorithm	Optimal	Worst	Mean	Standard
	SSA3	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
f_{11}	SSA	0.00E+00	5.04E-19	1.68E-20	9.21E-20
	SSA1	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	SSA2	0.00E+00	1.73E-261	5.78E-263	0.00E+00
	SSA3	0.00E+00	1.72E-23	5.74E-25	3.14E-24
	MSSA	0.00E+00	0.00E+00	0.00E+00	0.00E+00
f_{12}	SSA	-1.03E+00	1.88E-121	-9.63E-01	2.62E-01
	SSA1	-1.03E+00	0.00E+00	-9.28E-01	3.15E-01
	SSA2	-1.03E+00	-1.03E+00	-1.03E+00	2.32E-11
	SSA3	-1.03E+00	-1.03E+00	-1.03E+00	6.45E-05
	MSSA	-1.03E+00	-1.03E+00	-1.03E+00	1.47E-05
f_{13}	SSA	3.98E-01	3.98E-01	3.98E-01	3.15E-05
	SSA1	3.98E-01	8.45E-01	4.13E-01	8.16E-02
	SSA2	3.98E-01	3.98E-01	3.98E-01	5.35E-07
	SSA3	3.98E-01	6.42E-01	4.23E-01	6.90E-02
	MSSA	3.98E-01	3.98E-01	3.98E-01	7.41E-05
f_{14}	SSA	-3.86E+00	-3.09E+00	-3.75E+00	2.00E-01
	SSA1	-3.86E+00	-3.60E+00	-3.82E+00	5.91E-02
	SSA2	-3.86E+00	-3.09E+00	-3.81E+00	1.96E-01
	SSA3	-3.86E+00	-3.01E+00	-3.65E+00	2.03E-01
	MSSA	-3.86E+00	-3.86E+00	-3.86E+00	2.10E-03
f_{15}	SSA	-3.24E+00	-1.40E+00	-2.96E+00	3.62E-01
	SSA1	-3.30E+00	-2.82E+00	-3.10E+00	1.31E-01
	SSA2	-3.32E+00	-3.03E+00	-3.25E+00	9.02E-02
	SSA3	-3.17E+00	-1.17E+00	-2.65E+00	5.33E-01
	MSSA	-3.32E+00	-3.03E+00	-3.23E+00	9.01E-02
f_{16}	SSA	0.00E+00	1.63E-33	5.53E-35	2.97E-34
	SSA1	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	SSA2	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	SSA3	0.00E+00	4.17E-52	1.39E-53	7.62E-53
	MSSA	0.00E+00	0.00E+00	0.00E+00	0.00E+00

near the best position of the finder. The positions updated by Levy flight of the follower is changed from Eqs 16, 17:

$$x_{i,j}^{t+1} = \begin{cases} x_p^{t+1} + s \cdot \left|x_{x,j}^t - x_p^{t+1}\right| & if\ i > \dfrac{n}{2} \\ x_p^{t+1} + s \cdot \left|x_{x,j}^t - x_p^{t+1}\right| \cdot A^+ \cdot L & otherwise \end{cases} \tag{17}$$

Scouter position update strategy based on improved Cauchy mutations

In the SSA, the scouters enhance the global exploration ability of the algorithm to some extent, and the ability to jump out of the local optimal value is stronger if the number of scouters is a high percentage of the whole sparrow population. However, the random selection of scouters limits the more active sparrow individuals. The mechanism of fixing the number of scouter in the sparrow optimization algorithm slows down the optimization accuracy and convergence speed to a certain extent. Therefore, the improved scouter formula proposed in this paper on the original Eq. 11 is as follows:

$$x_{i,j}^{t+1} = \begin{cases} x_{i,j}^t + Q \cdot \left|x_{i,j}^t - x_{b,j}^t\right| & f_i \neq f_g \\ x_{b,j}^t + Q \cdot \left|x_w^t - x_{b,j}^t\right| & f_i = f_g \end{cases} \tag{18}$$

Because the sparrow optimization algorithm easily falls into a local optimum, the peak value of the Cauchy distribution function at the origin is small, but the distribution at both ends is relatively long. Because the range of the Cauchy

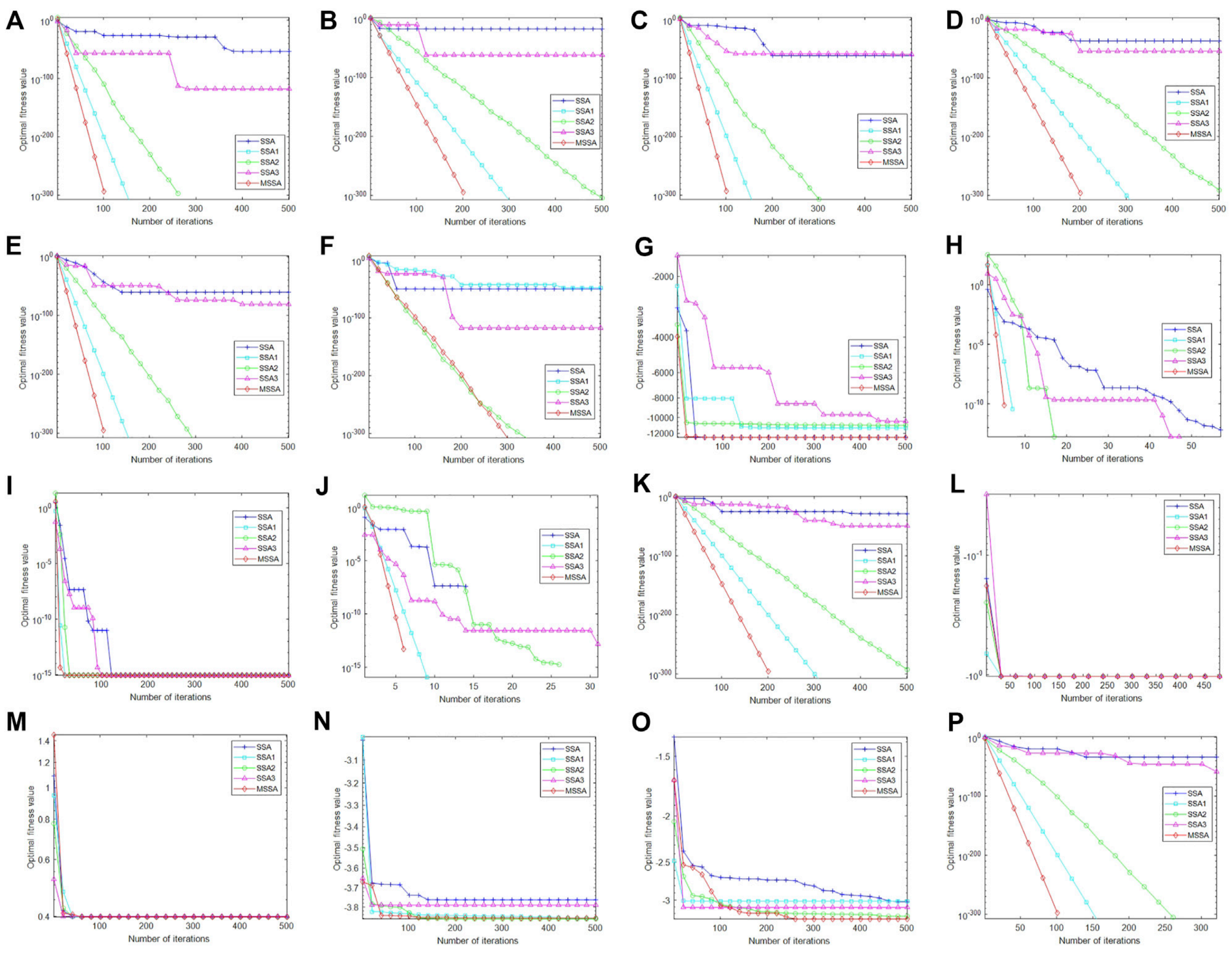

FIGURE 5
Mean convergence curve of the benchmark function. **(A)** f_1 mean convergence curve. **(B)** f2 mean convergence curve. **(C)** f3 mean convergence curve. **(D)** f4 mean convergence curve. **(E)** f5 mean convergence curve. **(F)** f6 mean convergence curve. **(G)** f7 mean convergence curve. **(H)** f8 mean convergence curve. **(I)** f9 mean convergence curve. **(J)** f10 mean convergence curve. **(K)** f11 mean convergence curve. **(L)** f12 mean convergence curve. **(M)** f13 mean convergence curve. **(N)** f14 mean convergence curve. **(O)** f15 mean convergence curve. **(P)** f_{16} mean convergence curve.

distribution function is relatively wide, it is easier to jump out of the local optimum by using the distribution at both ends of the Cauchy variation (Li et al., 2017). We used the Cauchy mutation to generate larger disturbances near the currently mutated sparrow individual to improve the local search ability of the algorithm and expand the search space of the algorithm. The standard Cauchy distribution function formula is as follows:

$$f_c(x) = \frac{1}{\pi}\left(\frac{\gamma}{(x - x_0)^2 + \gamma^2}\right) - \infty < x < \infty \tag{19}$$

If the random variable x of the Cauchy distribution obeys the position parameter of x_0 and the scale parameter of γ, it is recorded as Cauchy(γ, x_0). When the special case of $\gamma = 1$ and $x_0 = 0$, it becomes the standard Cauchy distribution function, and the corresponding cumulative distribution function is shown in Eq. 20:

$$F_c(x) = \frac{1}{2} + \frac{1}{\pi}\arctan\left(\frac{x - x_0}{\gamma}\right) \tag{20}$$

There are two main differences because the density function of the Cauchy distribution is similar to the Gaussian density function. On the one hand, the Cauchy distribution in the vertical direction is slightly smaller than the Gaussian distribution. On the other hand, the closer the Cauchy distribution is to the horizontal axis in the horizontal direction, the slower the change, so the Cauchy distribution can be regarded as infinite.

The comparison of the two distributions is shown in Figure 4. The Cauchy distribution and the Gaussian distribution have

TABLE 3 Comparison with the results of five metaheuristic algorithms.

Fun no.	Name	30 *dim*		200 *dim*		500 *dim*	
		Mean	SD	Mean	SD	Mean	SD
f_1	SSA[26]	1.28E-52	6.76E-52	5.12E-40	2.80E-39	4.12E-64	2.25E-63
	GWO[16]	2.12E-27	3.08E-27	1.17E-07	7.25E-08	1.66E-03	5.71E-04
	WOA[34]	2.21E-73	9.78E-73	4.55E-67	2.49E-66	2.68E-71	9.22E-71
	TSA[12]	1.91E-21	7.58E-21	3.64E-06	3.57E-06	2.77E-02	1.82E-02
	EO[35]	2.48E-41	3.99E-41	1.44E-25	2.04E-25	1.11E-22	1.64E-22
	MSSA	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
f_2	SSA[26]	3.12E-20	1.71E-19	3.18E-19	1.74E-18	1.89E-26	1.04E-25
	GWO[16]	1.18E-16	9.88E-17	3.26E-05	7.11E-06	1.21E-02	2.11E-03
	WOA[34]	3.09E-51	1.09E-50	6.10E-49	2.18E-48	6.87E-49	3.18E-48
	TSA[12]	9.65E-14	1.78E-13	5.24E-05	3.37E-05	6.84E-03	4.31E-03
	EO[35]	5.63E-24	4.83E-24	1.56E-15	8.88E-16	8.77E-14	6.13E-14
	MSSA	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
f_3	SSA[26]	7.58E-49	3.99E-48	1.04E-42	5.70E-42	2.66E-56	1.46E-55
	GWO[16]	1.79E-05	4.40E-05	2.02E+04	8.92E+03	3.22E+05	8.03E+04
	WOA[34]	4.52E+04	1.34E+04	5.17E+06	1.56E+06	3.02E+07	1.04E+07
	TSA[12]	1.22E-03	5.79E-03	1.72E+05	4.14E+04	1.42E+06	2.42E+05
	EO[35]	4.95E-09	5.06E-10	7.62E+02	1.53E+03	2.95E+04	3.08E+04
	MSSA	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
f_4	SSA[26]	7.37E-20	4.04E-19	1.41E-23	7.74E-23	3.32E-34	1.82E-33
	GWO[16]	9.14E-07	6.67E-07	2.38E+01	7.22E+00	6.40E+01	5.40E+00
	WOA[34]	5.07E+01	2.83E+01	7.59E+01	2.58E+01	8.04E+01	2.32E+01
	TSA[12]	4.57E-01	6.01E-01	9.39E+01	4.14E+00	9.91E+01	3.10E-01
	EO[35]	5.06E-10	9.39E-10	2.23E+01	2.48E+01	6.81E+01	1.59E+01
	MSSA	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
f_5	SSA[26]	1.72E-51	9.40E-51	4.54E-49	2.49E-48	8.45E-50	4.63E-49
	GWO[16]	4.03E-29	5.75E-29	1.68E-08	8.39E-09	8.45E-04	2.78E-04
	WOA[34]	1.56E-72	8.53E-72	3.49E-71	1.19E-70	1.70E-68	6.15E-68
	TSA[12]	3.22E-23	3.46E-23	1.07E-06	1.60E-06	1.32E-02	1.52E-02
	EO[35]	1.90E-42	3.16E-42	4.63E-26	1.30E-25	9.77E-23	9.61E-23
	MSSA	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
f_6	SSA[26]	1.93E-38	1.06E-37	3.33E-40	1.82E-39	1.38E-29	7.56E-29
	GWO[16]	2.59E-07	7.08E-07	9.92E+02	2.21E+02	3.88E+03	5.58E+02
	WOA[34]	4.98E+02	1.34E+02	3.33E+03	3.94E+02	8.03E+03	1.06E+03
	TSA[12]	2.79E-10	3.32E-10	5.03E+02	1.04E+02	1.93E+03	3.31E+02
	EO[35]	1.46E-05	3.12E-05	9.35E+02	3.08E+02	3.08E+03	9.19E+02
	MSSA	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
f_7	SSA[26]	-1.13E+04	1.90E+03	-7.62E+04	8.05E+03	-1.84E+05	2.15E+04
	GWO[16]	-6.15E+03	8.82E+02	-2.70E+04	6.46E+03	-5.67E+04	9.18E+03
	WOA[34]	-1.05E+04	1.79E+03	-6.86E+04	1.12E+04	-1.75E+05	2.84E+04
	TSA[12]	-5.96E+03	5.65E+02	-1.94E+04	1.19E+03	-3.09E+04	1.95E+03
	EO[35]	-8.85E+03	6.65E+02	-4.26E+04	2.74E+03	-7.41E+04	4.79E+03
	MSSA	-1.26E+04	9.57E-02	-8.38E+04	1.95E-01	-2.09E+05	1.12E+00
f_8	SSA[26]	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	GWO[16]	2.30E+00	2.95E+00	2.55E+01	1.45E+01	7.42E+01	2.30E+01
	WOA[34]	0.00E+00	0.00E+00	1.52E-14	8.30E-14	3.03E-14	1.66E-13
	TSA[12]	1.82E+02	3.73E+01	2.19E+03	2.31E+02	5.89E+03	5.98E+02

(Continued on following page)

TABLE 3 (*Continued*) Comparison with the results of five metaheuristic algorithms.

Fun no.	Name	30 *dim*		200 *dim*		500 *dim*	
		Mean	SD	Mean	SD	Mean	SD
	EO[35]	0.00E+00	0.00E+00	1.52E-14	5.77E-14	6.06E-14	2.31E-13
	MSSA	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
f_9	SSA[26]	8.88E-16	0.00E+00	8.88E-16	0.00E+00	8.88E-16	0.00E+00
	GWO[16]	1.11E-13	1.57E-14	2.42E-05	6.94E-06	1.95E-03	3.93E-04
	WOA[34]	5.27E-15	2.02E-15	5.27E-15	2.41E-15	3.85E-15	2.48E-15
	TSA[12]	1.87E+00	1.57E+00	2.72E-04	1.74E-04	1.06E-02	5.51E-03
	EO[35]	8.70E-15	2.17E-15	9.49E-14	2.87E-14	4.85E-13	2.46E-13
	MSSA	8.88E-16	0.00E+00	8.88E-16	0.00E+00	8.88E-16	0.00E+00
f_{10}	SSA[26]	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	GWO[16]	4.22E-03	1.01E-02	6.69E-03	1.43E-02	6.75E-03	2.50E-02
	WOA[34]	0.00E+00	0.00E+00	0.00E+00	0.00E+00	3.70E-18	2.03E-17
	TSA[12]	1.39E-02	1.98E-02	2.63E-02	5.94E-02	3.74E-02	7.96E-02
	EO[35]	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.11E-03	6.07E-03
	MSSA	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
f_{11}	SSA[26]	1.95E-33	1.06E-32	7.03E-24	3.20E-23	5.12E-20	2.81E-19
	GWO[16]	4.46E-04	6.57E-04	1.30E-02	3.27E-03	6.83E-02	1.38E-02
	WOA[34]	2.53E-43	1.38E-42	1.20E-49	4.86E-49	2.45E-50	7.46E-50
	TSA[12]	2.58E+01	6.50E+00	3.50E+02	4.66E+01	8.75E+02	1.96E+02
	EO[35]	1.41E-07	7.17E-07	1.23E-15	5.16E-15	1.19E-14	5.56E-15
	MSSA	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
f_{12}	SSA[26]	-1.03E+00	2.40E-07	-1.01E+00	1.45E-01	-8.60E-01	3.91E-01
	GWO[16]	-1.03E+00	1.82E-08	-1.03E+00	1.03E-08	-1.03E+00	6.06E-09
	WOA[34]	-1.03E+00	6.71E-10	-1.03E+00	6.26E-09	-1.03E+00	6.69E-10
	TSA[12]	-1.03E+00	8.02E-03	-1.03E+00	1.09E-02	-1.03E+00	5.77E-03
	EO[35]	-1.03E+00	6.39E-16	-1.03E+00	6.18E-16	-1.03E+00	6.39E-16
	MSSA	-1.03E+00	1.46E-05	-1.03E+00	1.31E-05	-1.03E+00	1.46E-05
f_{13}	SSA[26]	3.98E-01	8.02E-06	3.98E-01	9.38E-06	3.98E-01	5.02E-06
	GWO[16]	3.98E-01	3.88E-05	3.98E-01	1.32E-06	3.98E-01	1.27E-04
	WOA[34]	3.98E-01	4.45E-05	3.98E-01	9.77E-06	3.98E-01	8.92E-06
	TSA[12]	3.98E-01	9.97E-05	3.98E-01	3.50E-05	3.98E-01	8.92E-05
	EO[35]	3.98E-01	0.00E+00	3.98E-01	0.00E+00	3.98E-01	0.00E+00
	MSSA	3.98E-01	9.24E-05	3.98E-01	5.85E-05	3.98E-01	6.99E-05
f_{14}	SSA[26]	-3.81E+00	6.70E-02	-3.76E+00	1.95E-01	-3.80E+00	1.47E-01
	GWO[16]	-3.86E+00	2.46E-03	-3.86E+00	2.60E-03	-3.86E+00	1.64E-03
	WOA[34]	-3.85E+00	1.92E-02	-3.86E+00	6.28E-03	-3.86E+00	1.07E-02
	TSA[12]	-3.86E+00	1.43E-03	-3.86E+00	1.41E-03	-3.86E+00	1.10E-04
	EO[35]	-3.86E+00	1.44E-03	-3.86E+00	2.49E-15	-3.86E+00	2.49E-15
	MSSA	-3.86E+00	2.03E-02	-3.86E+00	6.59E-03	-3.86E+00	4.01E-03
f_{15}	SSA[26]	-3.05E+00	2.17E-01	-2.96E+00	4.63E-01	-2.91E+00	3.37E-01
	GWO[16]	-3.25E+00	7.06E-02	-3.24E+00	7.51E-02	-3.27E+00	6.64E-02
	WOA[34]	-3.25E+00	9.10E-02	-3.20E+00	1.26E-01	-3.26E+00	8.98E-02
	TSA[12]	-3.24E+00	1.26E-01	-3.26E+00	7.44E-02	-3.25E+00	1.04E-01
	EO[35]	-3.24E+00	6.99E-02	-3.28E+00	5.85E-02	-3.25E+00	6.92E-02
	MSSA	-3.21E+00	8.80E-02	-3.24E+00	7.45E-02	-3.21E+00	9.89E-02
f_{16}	SSA[26]	4.77E-40	2.61E-39	1.48E-59	8.09E-59	3.49E-64	1.91E-63
	GWO[16]	9.42E-207	0.00E+00	8.88E-205	0.00E+00	1.00E-210	0.00E+00

(Continued on following page)

TABLE 3 (*Continued*) Comparison with the results of five metaheuristic algorithms.

Fun no.	Name	30 *dim*		200 *dim*		500 *dim*	
		Mean	SD	Mean	SD	Mean	SD
	WOA[34]	4.18E-111	2.29E-110	4.90E-106	2.68E-105	1.38E-109	7.56E-109
	TSA[12]	6.90E-95	3.78E-94	1.47E-94	8.04E-94	7.55E-105	4.13E-104
	EO[35]	3.42E-205	0.00E+00	4.86E-213	0.00E+00	4.10E-208	0.00E+00
	MSSA	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00

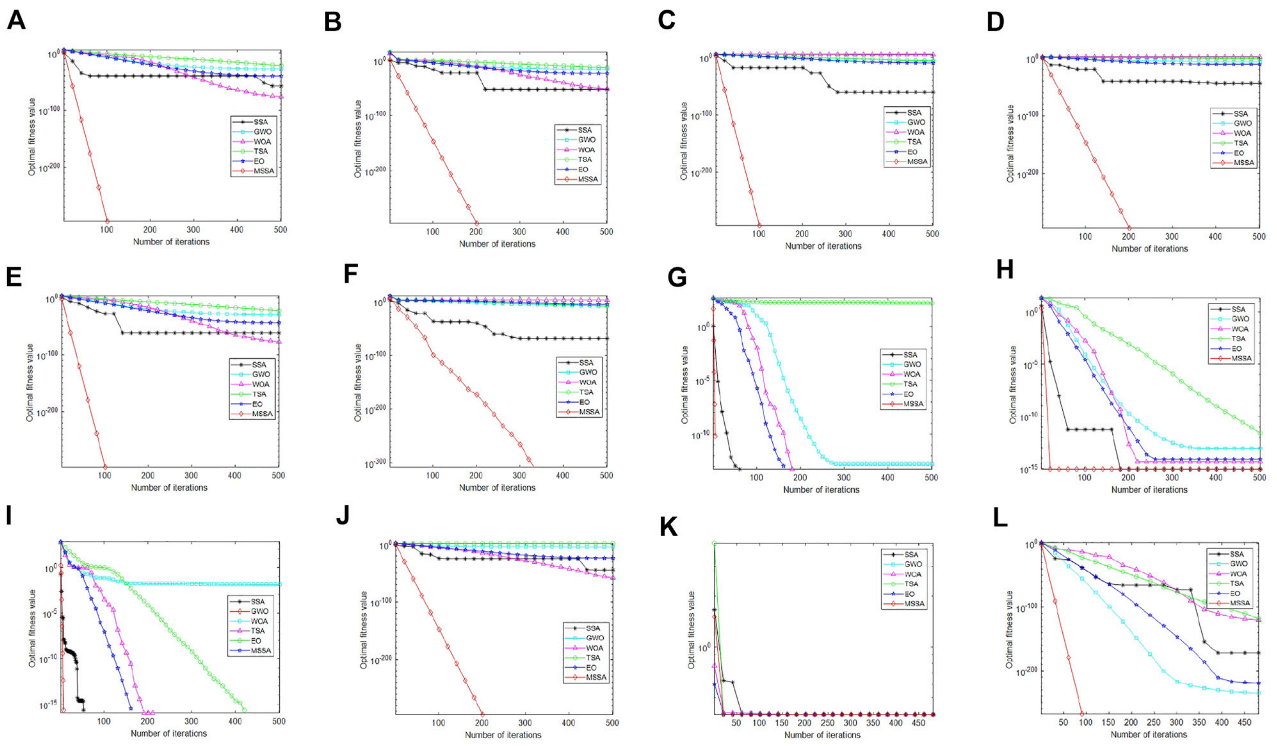

FIGURE 6
Comparison of optimization of each algorithm. **(A)** f1 mean convergence curve. **(B)** f2 mean convergence curve. **(C)** f3 mean convergence curve. **(D)** f4 mean convergence curve. **(E)** f5 mean convergence curve. **(F)** f6 mean convergence curve. **(G)** f8 mean convergence curve. **(H)** f9 mean convergence curve. **(I)** f10 mean convergence curve. **(J)** f11 mean convergence curve. **(K)** f13 mean convergence curve. **(L)** f_{16} mean convergence curve.

certain similarities and their own characteristics. The Cauchy distribution has a wider range than the Gaussian distribution. The Cauchy distribution easily generates a random number far from the origin, which means that the sparrow individual after Cauchy mutation has a higher probability of jumping out of the local optimal area. In addition, the peak value of the Cauchy distribution function is lower than that of the Gaussian distribution, which can shorten the search time of the mutated sparrow individuals around the neighborhood. Therefore, this paper integrates the Cauchy mutation to improve the scouter position update strategy, increases the diversity of the population and improves the ability of the algorithm to jump out of the local optimum. The position update formula is as follows:

$$x_{i,j}^{t+1} = x_{best}^{t} + x_{i,j}^{t} \cdot Cauchy\,(0,1) \tag{21}$$

Eqs 9–11, derived from the original work of SSA, specify how the sparrow is updated and construct the basic flow of the

TABLE 4 Wilcoxon rank sum test results.

Fun. No	SSA(p_1)	GWO(p_2)	WOA(p_3)	TSA(p_4)	EO(p_5)
f_1	1.31×10^{-07}	1.21 × 10–12	1.21 × 10–12	1.21 × 10–12	1.21 × 10–12
f_2	1.93×10^{-10}	1.21 × 10–12	1.21 × 10–12	1.21 × 10–12	1.21 × 10–12
f_3	1.70×10^{-08}	1.21 × 10–12	1.21 × 10–12	1.21 × 10–12	1.21 × 10–12
f_4	1.66×10^{-11}	1.21 × 10–12	1.21 × 10–12	1.21 × 10–12	1.21 × 10–12
f_5	1.31×10^{-07}	1.21 × 10–12	1.21 × 10–12	1.21 × 10–12	1.21 × 10–12
f_6	1.70×10^{-08}	1.21 × 10–12	1.21 × 10–12	1.21 × 10–12	1.21 × 10–12
f_7	2.15×10^{-10}	3.02 × 10–11	3.34 × 10–11	3.02 × 10–11	3.02 × 10–11
f_8	NaN	1.17 × 10–12	NaN	1.21 × 10–12	NaN
f_9	NaN	1.15 × 10–12	3.06 × 10–09	1.21 × 10–12	3.13 × 10–12
f_{10}	NaN	6.62 × 10–04	3.04 × 10–01	8.86 × 10–07	NaN
f_{11}	1.93×10^{-10}	1.21 × 10–12	1.21 × 10–12	1.21 × 10–12	1.21 × 10–12
f_{12}	3.34×10^{-11}	3.02 × 10–11	3.02 × 10–11	2.13 × 10–05	6.32 × 10–12
f_{13}	3.83×10^{-06}	1.07 × 10–09	5.00 × 10–09	8.31 × 10–03	1.21 × 10–11
f_{14}	6.41×10^{-01}	3.77 × 10–04	5.59 × 10–01	5.53 × 10–08	1.72 × 10–12
f_{15}	1.03×10^{-06}	9.79 × 10–05	6.55 × 10–04	3.34 × 10–03	3.06 × 10–11
f_{16}	1.31×10^{-07}	1.21 × 10–12	1.21 × 10–12	1.21 × 10–12	1.21 × 10–12
+/ = /-	13/3/3	16/0/0	15/1/1	16/0/0	14/2/2

algorithm. Based on the idealization and feasibility of the above model, the basic steps of the improved SSA can be summarized in the pseudo-code shown in Algorithm 1.

Simulation experiments of MSSA

Experimental environment parameters and test functions

The computer configuration used in the simulation experiment is Intel Core i7-6700HQ, the main frequency is 2.60 GHz, 8 GB memory, 64 bit operating system, and the computing environment is MATLAB 2019(b). We compare the SSA, GWO, WOA (Mirjalili and Lewis, 2016), TSA, and EO (Faramarzi et al., 2020) with MSSA. The basic parameters of the algorithms were set to the same value, including population size $N = 30$, the maximum number of iterations $T_{\max} = 500$. The test function dimensions were divided into low-dimensional ($d = 30$) and high-dimensional ($d = 200$ and $d = 500$).

To test the optimization performance of the MSSA algorithm, 16 benchmark functions with different characteristics used in the literature were selected for the function optimization test. The selected test functions were divided into three categories. The first category was the single-modal test function, f_1-f_6, which is mainly used to evaluate the optimization accuracy and convergence speed of the algorithm. The second type is the multi-modal test function, f_7-f_{11}, which is used to test the exploration ability of the algorithm and the ability to jump out of the local optimal value. The third type is the fixed multi-modal test function, f_{12}-f_{16}. This paper used the 16 benchmark functions to evaluate the comprehensive ability of the algorithm. The detailed description and related information are shown in Table 1.

Comparing MSSA with various improvement strategies

Optimization performance experiment

To fully verify the optimization effect of the proposed improved strategy, we denoted the three improvement strategies as SSA1, SSA2, and SSA3, and compare them with SSA and MSSA. Each algorithm was independently run 30 times on 16 benchmark functions, and the optimal value, worst value, mean value, and standard value were recorded. The optimal value and the worst value reflected the single optimization ability of the algorithm, the average value reflected the convergence accuracy, and the standard deviation reflected the stability and robustness of the algorithm. Parameters were uniformly set as follows: population size $N = 30$, search space dimension dim = 30, and maximum number of iterations, $T_{\max} = 500$. The results of experimental optimization are shown in Table 2:

Table 2 shows that for the single-modal test functions, f_1-f_6, MSSA can find the theoretical optimal value, illustrating that the stability is strong. At the same time, SSA1 can reach the theoretical optimal value and the more stable standard, showing that the introduction of the lens imaging learning strategy helps the algorithm to jump out of the local optimal

TABLE 5 Part of the CEC2014 function.

Fun no.	Dim	Function type	Range	Optimal
CEC01	30	UN	[-100,100]	100
CEC02	30	UN	[-100,100]	200
CEC03	30	UN	[-100,100]	300
CEC04	30	MF	[-100,100]	400
CEC05	30	MF	[-100,100]	500
CEC06	30	MF	[-100,100]	600
CEC07	30	MF	[-100,100]	700
CEC08	30	MF	[-100,100]	800
CEC09	30	MF	[-100,100]	900
CEC10	30	MF	[-100,100]	1000
CEC11	30	MF	[-100,100]	1100
CEC12	30	MF	[-100,100]	1200
CEC13	30	MF	[-100,100]	1300
CEC14	30	MF	[-100,100]	1400
CEC15	30	MF	[-100,100]	1500
CEC16	30	MF	[-100,100]	1600
CEC17	30	HF	[-100,100]	1700
CEC18	30	HF	[-100,100]	1800
CEC19	30	HF	[-100,100]	1900
CEC20	30	HF	[-100,100]	2000
CEC21	30	HF	[-100,100]	2100
CEC22	30	HF	[-100,100]	2200
CEC23	30	HF	[-100,100]	2300
CEC24	30	HF	[-100,100]	2400
CEC25	30	CF	[-100,100]	2500
CEC26	30	CF	[-100,100]	2600
CEC27	30	CF	[-100,100]	2700
CEC28	30	CF	[-100,100]	2800
CEC29	30	CF	[-100,100]	2900
CEC30	30	CF	[-100,100]	3000

value. SSA2 can reach the optimal theoretical value in f_1, f_2, f_5, and f_6, indicating that adding Levy flight can help the population to deeply mine the global optimal values and improve the global search ability of the algorithm. For the multi-modal test functions, f_7-f_{11}, there are a large number of local extremes distributed in their solution space, and it is difficult for the algorithm to perform global optimization. The MSSA and other comparative algorithms fall into local optimal solutions when solving function f_9. But MSSA has higher convergence accuracy and stability in other test functions, and the standards are more stable than other algorithms.

For the fixed multi-modal functions, f_{12}-f_{16}, MSSA can find the theoretical optimal value, and the stability is extremely strong and the standard value is lower than SSA1, SSA2, and SSA3, indicating that MSSA has stronger stability and robustness.

Convergence curve experiment

We used the average convergence curve to reflect the dynamic convergence characteristics of the MSSA in this paper. We make the algorithm run 30 times independently under the population size $N = 30$, maximum number of iterations $T_{\max} = 500$, and the search dimension dim = 30. Figure 5 presents the average convergence curves of the 16 benchmark functions.

Figure 5 shows that the MSSA has a higher optimization accuracy solution rate and faster convergence speed in f_1-f_5 than under the same number of iterations. The values of f_7-f_{11} and f_{16} show that the MSSA is able to guarantee the exploration ability and illustrate that the MSSA can ensure the development ability without losing the population diversity and optimization stability. For f_6 and f_{12}-f_{14}, the MSSA can converge to the optimal value faster in the later stages, which indicates that adopting the three strategies can help the algorithm jump out of the local optimal value effectively. In general, the MSSA average convergence curve is below the four comparison algorithms and takes fewer iterations to reach the theoretical optimal. From Table 2 and Figure 5, the result illustrates that the MSSA has higher convergence accuracy, convergence speed, stronger stability, and better robustness. The experimental results verify the effectiveness of the proposed algorithm and achieve the purpose of improving the standard SSA.

Comparing MSSA with other algorithms

Optimization performance experiment

To further test the optimization characteristics of the MSSA algorithm for the benchmark function, this paper chose the standard SSA, GWO, WOA, TSA, EO, and MSSA to compare for optimization performance. For each benchmark function, the search dimension is set to dim = 30/200/500, the maximum number of iterations $T_{\max} = 500$ and the population size is $N = 30$. The test functions of Table 1 are used to perform the optimization comparison test, and each algorithm is run 30 times independently. The comparison results are shown in Table 3:

As seen from Table 3, the solution rate of MSSA can reach 100% when solving for the single-modal test function, f_1-f_6, which indicates that MSSA has good optimization accuracy and robustness. For the multi-modal test function, f_7-f_{11}, MSSA can find the theoretical optimal value. For the fixed multi-modal test function, f_{12}-f_{16}, the MSSA solution results are very close to or equal to the theoretical optimal value. As the dimension of the search space increases from 30 to 200 and 500 dimensions, the search accuracy and stability of the algorithm decrease because the optimization process requires more computations, but MSSA still has the highest optimization accuracy. This illustrates that the MSSA has a significant

TABLE 6 CEC2014 function optimization comparison.

No.	SSA		GWO		WOA		TSA		EO		MSSA	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	Std
CEC01	9.554E+08	2.631E+08	1.098E+08	8.610E+07	2.175E+08	1.105E+08	4.116E+08	2.529E+08	1.607E+07	7.842E+06	7.625E+08	2.612E+08
CEC02	7.477E+10	8.241E+09	3.206E+09	2.525E+09	6.849E+09	2.531E+09	3.020E+10	1.016E+10	1.450E+05	1.666E+05	5.933E+10	8.122E+09
CEC03	8.759E+04	3.076E+03	5.134E+04	1.349E+04	1.364E+05	7.675E+04	5.817E+04	1.096E+04	2.289E+04	1.232E+04	8.518E+04	4.116E+03
CEC04	1.369E+04	3.074E+03	6.920E+02	8.595E+01	1.350E+03	3.125E+02	3.325E+03	2.152E+03	5.261E+02	3.970E+01	9.841E+03	2.116E+03
CEC05	5.212E+02	9.680E-02	5.211E+02	7.219E-02	5.209E+02	1.063E-01	5.211E+02	7.515E-02	5.210E+02	8.605E-02	5.210E+02	7.811E-02
CEC06	6.432E+02	2.736E+00	6.169E+02	2.231E+00	6.388E+02	3.369E+00	6.331E+02	3.501E+00	6.112E+02	2.621E+00	6.405E+02	2.563E+00
CEC07	1.258E+03	1.513E+02	7.296E+02	2.719E+01	7.462E+02	1.945E+01	9.799E+02	8.968E+01	7.002E+02	1.168E-01	1.105E+03	9.123E+01
CEC08	1.134E+03	2.557E+01	9.024E+02	3.450E+01	1.052E+03	5.699E+01	1.072E+03	3.917E+01	8.621E+02	1.661E+01	1.099E+03	3.076E+01
CEC09	1.280E+03	3.438E+01	1.023E+03	3.495E+01	1.218E+03	5.999E+01	1.228E+03	4.629E+01	1.004E+03	2.524E+01	1.226E+03	2.770E+01
CEC10	8.566E+03	6.372E+02	3.904E+03	1.268E+03	6.099E+03	9.402E+02	6.759E+03	8.883E+02	3.408E+03	6.872E+02	7.815E+03	5.461E+02
CEC11	8.764E+03	6.220E+02	5.338E+03	1.774E+03	7.636E+03	4.399E+02	7.387E+03	6.106E+02	5.206E+03	7.782E+02	8.437E+03	5.112E+02
CEC12	1.204E+03	1.267E+00	1.203E+03	1.223E+00	1.202E+03	5.587E-01	1.203E+03	4.669E-01	1.202E+03	5.199E-01	1.203E+03	6.070E-01
CEC13	1.308E+03	9.376E-01	1.301E+03	2.225E-01	1.302E+03	9.442E-01	1.304E+03	7.534E-01	1.300E+03	1.018E-01	1.307E+03	7.269E-01
CEC14	1.608E+03	3.762E+01	1.406E+03	7.359E+00	1.420E+03	1.042E+01	1.505E+03	2.852E+01	1.400E+03	1.942E-01	1.578E+03	2.299E+01
CEC15	1.105E+05	4.523E+04	1.804E+03	5.126E+02	2.550E+03	1.105E+03	2.256E+04	3.623E+04	1.510E+03	2.472E+00	5.588E+04	2.556E+04
CEC16	1.613E+03	3.680E-01	1.612E+03	4.505E-01	1.613E+03	4.879E-01	1.613E+03	3.369E-01	1.612E+03	6.038E-01	1.613E+03	2.314E-01
CEC17	1.063E+08	9.442E+07	3.127E+06	2.652E+06	2.355E+07	1.383E+07	1.336E+07	1.277E+07	1.116E+06	7.946E+05	9.002E+07	5.939E+07
CEC18	3.101E+09	2.708E+09	2.865E+07	3.139E+07	6.831E+06	6.326E+06	6.929E+08	1.498E+09	6.086E+03	1.168E+04	1.666E+09	1.164E+09
CEC19	2.377E+03	1.769E+02	1.959E+03	2.991E+01	2.020E+03	6.305E+01	2.090E+03	9.015E+01	1.915E+03	1.975E+01	2.230E+03	1.034E+02
CEC20	6.555E+05	1.016E+06	3.720E+04	1.914E+04	1.578E+05	1.533E+05	7.574E+04	5.123E+04	2.100E+04	9.639E+03	2.980E+05	2.909E+05
CEC21	5.064E+07	4.092E+07	2.340E+06	3.394E+06	1.183E+07	9.843E+06	6.214E+06	6.954E+06	5.188E+05	3.946E+05	3.479E+07	3.409E+07
CEC22	5.127E+03	1.861E+03	2.708E+03	1.610E+02	3.238E+03	2.960E+02	3.487E+03	8.483E+02	2.608E+03	2.130E+02	4.939E+03	5.236E+03
CEC23	2.500E+03	0.000E+00	2.641E+03	1.343E+01	2.715E+03	3.445E+01	2.740E+03	9.350E+01	2.615E+03	1.493E-01	2.500E+03	0.000E+00
CEC24	2.600E+03	0.000E+00	2.600E+03	3.613E-02	2.610E+03	6.295E+00	2.610E+03	2.017E+01	2.600E+03	7.775E-03	2.600E+03	0.000E+00
CEC25	2.700E+03	0.000E+00	2.713E+03	5.881E+00	2.718E+03	2.124E+01	2.730E+03	1.082E+01	2.701E+03	2.503E+00	2.700E+03	0.000E+00
CEC26	2.769E+03	4.420E+01	2.747E+03	5.048E+01	2.717E+03	3.770E+01	2.806E+03	6.982E+01	2.727E+03	4.479E+01	2.766E+03	4.594E+01
CEC27	2.900E+03	0.000E+00	3.422E+03	1.145E+02	3.924E+03	3.295E+02	3.867E+03	2.875E+02	3.282E+03	1.095E+02	2.900E+03	0.000E+00
CEC28	3.000E+03	0.000E+00	4.327E+03	5.250E+02	5.274E+03	7.162E+02	7.836E+03	9.600E+02	3.777E+03	1.204E+02	3.000E+03	0.000E+00
CEC29	3.100E+03	0.000E+00	3.670E+06	5.556E+06	1.800E+07	1.944E+07	8.854E+07	7.142E+07	3.425E+06	4.721E+06	3.100E+03	0.000E+00
CEC30	3.200E+03	0.000E+00	9.243E+04	6.681E+04	5.368E+05	3.128E+05	6.287E+05	3.630E+05	1.109E+04	7.519E+03	3.200E+03	0.000E+00

competitive advantage and stability in solving high-dimensional and complex optimization problems.

High-dimensional convergence curve experiment

To compare the optimization performance of the MSSA and other algorithms, we selected SSA, GWO, WOA, TSA, EO, and MSSA for high-dimensional function optimization comparison. This paper used the 12 representative benchmark functions given in Table 1, the dimension dim = 500, and the maximum number of iterations, $T_{\max}$ = 500. The high-dimensional optimization curve of each algorithm is shown in Figures 6A–L. The convergence accuracy and speed of the MSSA were significantly higher than those of the other algorithms, which indicate that the multi-strategy can effectively prevent the algorithm from falling into local optima. The results of the optimization curve illustrated that for the single-modal test functions, f_1-f_6, the multi-modal functions test functions, f_8, f_{10}, f_{11}, and the fixed multi-modal function, f_{16}, MSSA can find the optimal value faster with fewer iterations and higher convergence accuracy. For the multi-modal function, f_9, the optimal value of MSSA was similar to several other contrast functions, but the MSSA converged much faster. For the fixed multi-modal function, f_{13}, MSSA fell into a local optimum similar to the five compared algorithms, but it still could find the theoretical optimum; thus, MSSA has significant advantages over the other algorithms.

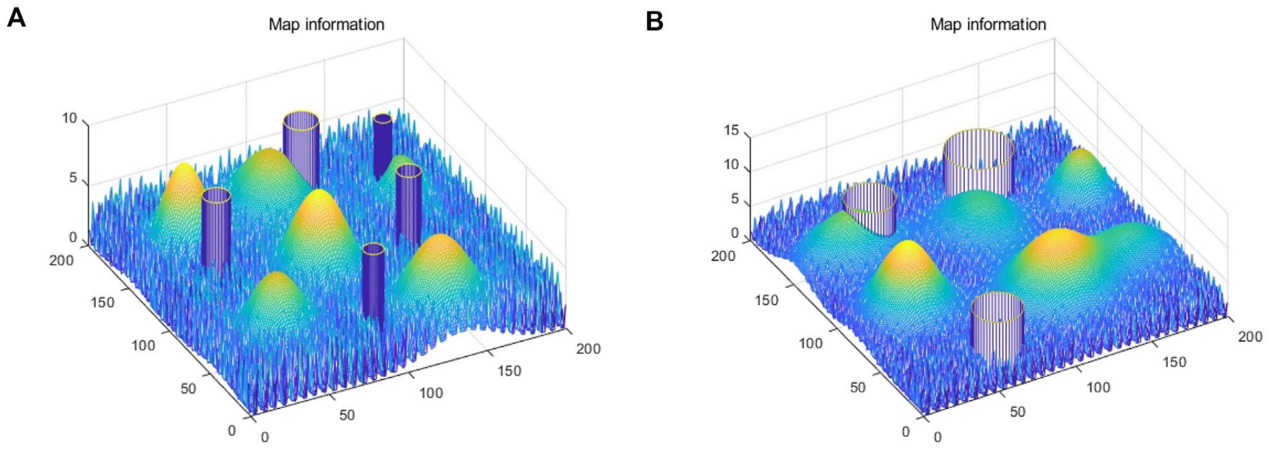

FIGURE 7
3D map model information of two environments. **(A)** 3D map of environment 1. **(B)** 3D map of environment 2.

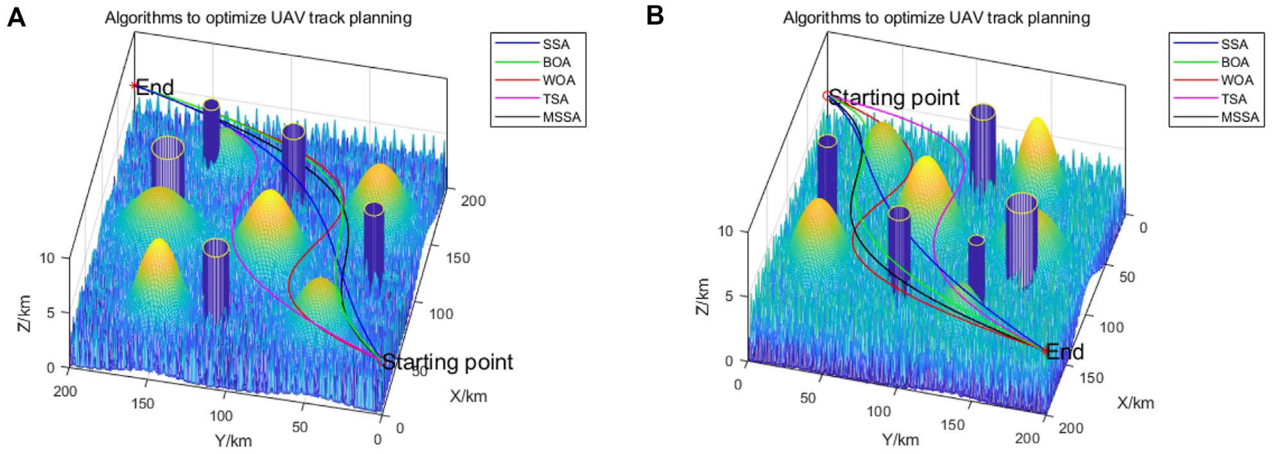

FIGURE 8
Simulation results of 3D trajectory planning in environment 1 from two different views.

Wilcoxon rank-sum test experiment

In general, only the mean and standard deviation are used for data analysis. The Wilcoxon rank-sum test is a non-parametric test method for the mean that is not limited by the overall distribution and has a wide range of applications (Derrac, J. et al., 2011). To more comprehensively analyze the performance difference between the MSSA and other algorithms, we selected the running results of MSSA and five comparison algorithms in sixteen test functions to conduct the Wilcoxon rank sum test. The p value can be considered to reject the null hypothesis when $p < 5\%$, indicating that there is a significant difference in the test results. *NaN* indicates that there are no data to compare with the algorithm, +, =, and - indicate that the MSSA algorithm's optimization performance is better than, equal to, and worse than the compared algorithms, respectively. The results of the Wilcoxon rank sum test are shown in Table 4:

As seen from Table 4, the MSSA p value of the Wilcoxon rank sum test is less than 5%. The experimental results indicate that there is a significant difference between the MSSA and the other five algorithms, and the MSSA is significantly better than the others, further reflecting the robustness of the MSSA.

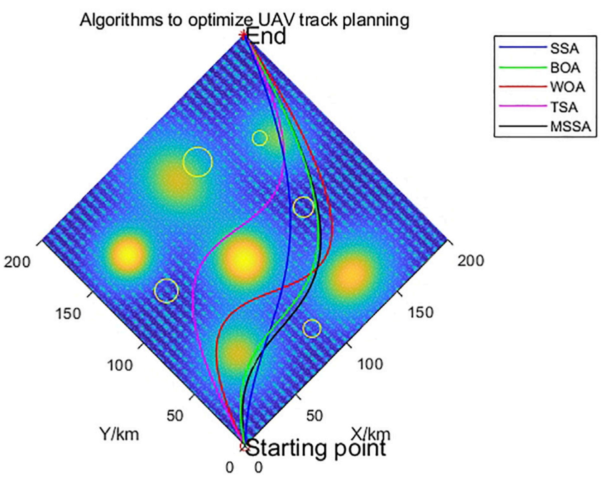

FIGURE 9
Comparison of 2D track planning in environment 1.

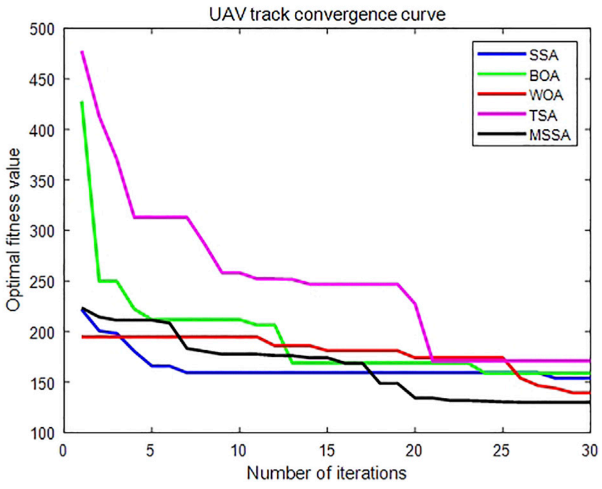

FIGURE 10
Convergence diagram of track cost function in environment 1.

CEC2014 test functions experiment

Most of the CEC2014 test functions (Tejani, 2018) are composed of the weights of multiple basic optimizations test functions, which makes the characteristics of the test functions more complex. In this paper, the proposed MSSA was tested against these complex test functions. On the one hand, these functions can effectively reflect the superior performance of MSSA for optimization of a complex function. On the other hand, the combinatorial optimization of multiple test functions reflects the applicability of SMSA to different complex optimization problems. To further test the performance of MSSA, this paper chose the CEC2014 single objective optimization function for solution analysis, including unit-modal, multi-modal, Hybrid, and composition type functions. Table 5 shows the relevant information of CEC2014 functions. This study compared the MSSA with five algorithms, including SSA, GWO, WOA, TSA, and EO. To ensure fairness of the algorithm comparison, the maximum number of iterations, $T_{\max} = 1000$, the population size was $N = 30$, and the dimension, dim = 30, were set to the same values in each algorithm, which were run 30 times independently, and the mean and standard deviation were recorded. The results are shown in Table 6.

```
The algorithm initialization phase:
$T$: the maximum iterations;
$Dd$: the number of discovers;
$Sd$: the number of scouters;
$R_2$: the alarm value;
$n$: the number of sparrows;
Perform sparrow population position initialization;
The algorithm iterates the search phase:
while $t < T_{\max}$;
    Ranking of individual fitness values to find the best and worst individuals;
    $R_2 = rand(1)$;
    for $i = 1 : Dd$;
        Using Eq. 14 update the sparrow's location;
    end for;
    for $i = 1 : (Dd+1)$;
        Using Eq. 17 update the sparrow's location;
    end for;
    for i = 1 : $Sd$;
        Using Eq. 21 update the sparrow's location;
    end for;
    Get the current optimal position;
    $t = t+1$;
end while;
Output global optimum;
```

Algorithm 1. The framework of the MSSA.

Table 6 shows that MSSA was better than SSA on 22 test functions, including CEC01, CEC02, CEC03, CEC04, CEC05, CEC06, CEC07, CEC08, CEC09, CEC10, CEC11, CEC12, CEC13, CEC14, CEC15, CEC17, CEC18, CEC19, CEC20, CEC21, CEC22, and CEC26. MSSA can find values close to SSA on CEC16, CEC23, CEC24, CEC25, CEC26, CEC28, CEC29, and CEC30. In terms of standard deviation, MSSA is better than SSA, GWO, and TSA on most test functions. It is worth noting that MSSA is better than SSA in the remaining 25 test functions except CEC03, CEC05, CEC16, CEC22, and CEC26. Generally,

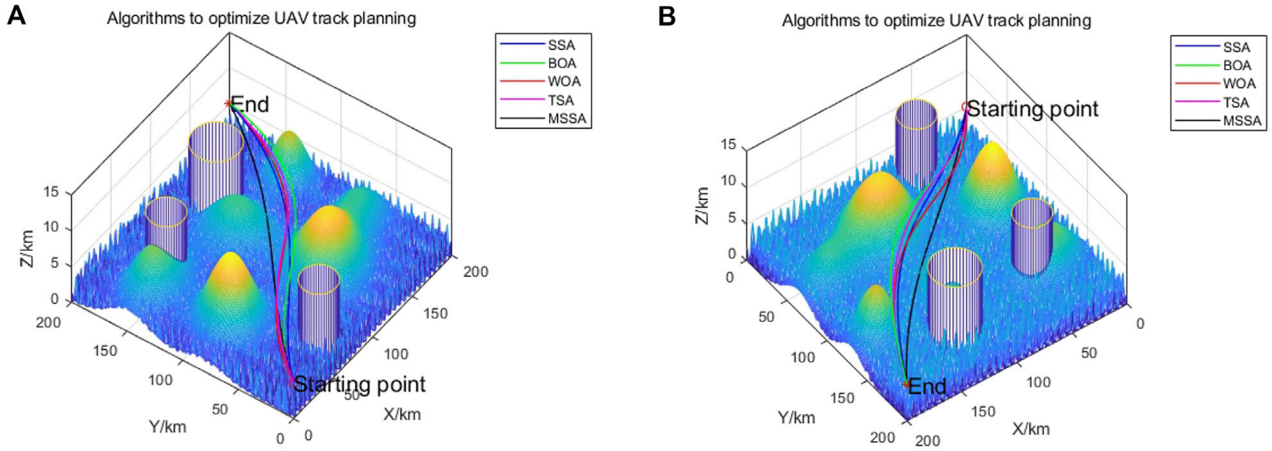

FIGURE 11
Simulation results of 3D trajectory planning in environment 2 from two different views.

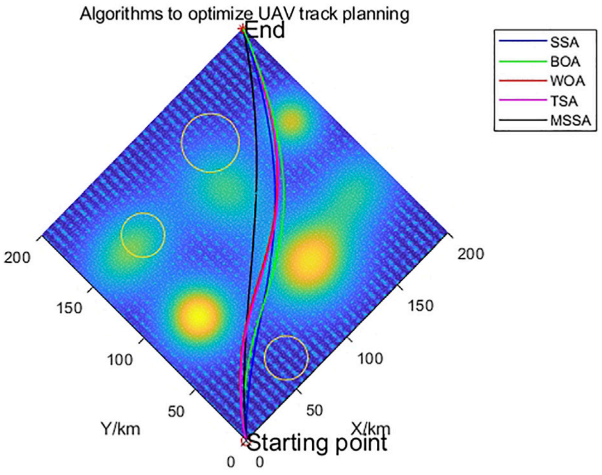

FIGURE 12
Comparison of 2D track planning in environment 2.

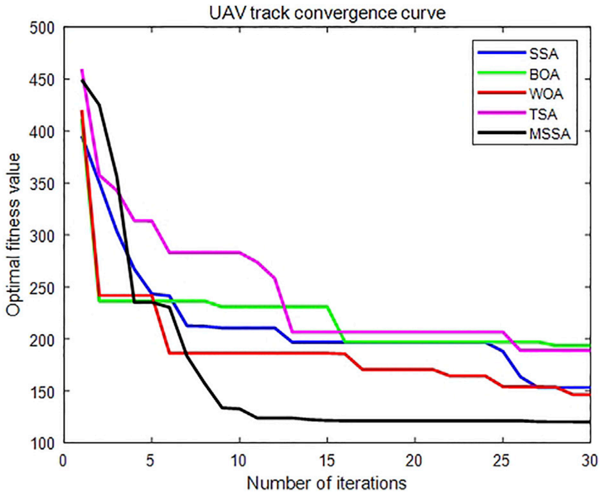

FIGURE 13
Convergence diagram of track cost function in environment 2.

the proposed algorithm has more prominent advantages in the CEC2014 test function compared with the other eight algorithms.

UAV trajectory planning based on MSSA

In this section, we established two mathematical models of trajectory planning constraints in MATLAB and simulation experiments were carried out in a 3-D environment. This experiment combines the content of the above sections, and we conducted simulation experiments on UAV trajectory planning in two challenging mission environments. We compared MSSA with the SSA, BOA, WOA, and TSA to verify the performance improvement of the MSSA in the trajectory planning problem.

Experimental simulation parameters

The first step in UAV trajectory planning was to initialize the relevant parameters. The basic parameters of SSA before and after the improvement were set as follows: the population size was $N = 50$, the maximum number of iterations was $T_{\max} = 30$,

TABLE 7 Experimental data of environment 1.

No.	SSA	BOA	WOA	TSA	MSSA
1	221.9858	428.1988	194.8475	477.7604	223.6669
2	200.8703	249.9676	194.8475	412.8379	214.5968
3	198.3777	249.9676	194.8475	370.6239	211.5184
4	180.7114	222.4805	194.8475	313.0749	211.5184
5	166.1277	211.9584	194.8475	313.0749	211.5184
6	166.1277	211.9584	194.8475	313.0749	208.6994
7	159.4911	211.9584	194.8475	313.0749	183.4661
8	159.4911	211.9584	194.8475	286.9255	180.6388
9	159.4911	211.9584	194.8475	258.2724	177.8031
10	159.4911	211.9584	194.8475	258.2724	177.8031
11	159.4911	206.6703	194.8475	252.3117	177.8031
12	159.4911	206.6703	186.2235	252.3117	176.5868
13	159.4911	169.1189	186.2235	251.7576	176.4338
14	159.4911	169.1189	186.2235	247.0899	174.2706
15	159.4911	169.1189	181.2497	247.0899	174.2706
16	159.4911	169.1189	181.2497	247.0899	168.9351
17	159.4911	169.1189	181.2497	247.0899	168.9351
18	159.4911	169.1189	181.2497	247.0899	148.9273
19	159.4911	169.1189	181.2497	247.0899	148.9273
20	159.4911	169.1189	174.2521	171.1681	134.4919
21	159.4911	169.1189	174.2521	171.1681	134.4919
22	159.4911	169.1189	174.2521	171.1681	132.033
23	159.4911	169.1189	174.2521	171.1681	132.033
24	159.4911	158.9664	174.2521	171.1681	131.4209
25	159.4911	158.9664	174.2521	171.1681	130.6637
26	159.4911	158.9664	154.1683	171.1681	130.3178
27	159.4911	158.9664	146.8336	171.1681	130.3178
28	154.0068	158.9664	144.2572	171.1681	130.3178
29	154.0068	158.9664	139.5953	171.1681	130.3178
30	154.0068	158.9664	139.5953	171.1681	130.26

TABLE 8 Experimental data of environment 2.

No.	SSA	BOA	WOA	TSA	MSSA
1	395.4108	411.9735	420.1169	459.6827	449.4203
2	350.7416	236.5109	241.8295	357.607	425.1031
3	303.5717	236.5109	241.8295	342.8342	356.2115
4	267.1788	236.5109	241.8295	313.7257	235.2283
5	243.3833	236.5109	241.8295	313.5049	235.2283
6	241.4834	236.5109	186.2959	283.0012	230.3628
7	212.6503	236.5109	186.2959	283.0012	183.3411
8	212.1987	236.5109	186.2959	283.0012	157.4026
9	210.5834	230.8149	186.2959	283.0012	133.7521
10	210.5834	230.8149	186.2959	283.0012	132.6643
11	210.5834	230.8149	186.2959	273.8353	123.9401
12	210.5834	230.8149	186.2959	258.2411	123.9401
13	196.827	230.8149	186.2959	206.6917	123.9401
14	196.827	230.8149	186.2959	206.6917	122.3308
15	196.827	230.8149	186.2959	206.6917	121.5816
16	196.827	197.1322	185.5324	206.6917	121.3304
17	196.827	197.1322	170.6196	206.6917	121.3304
18	196.827	197.1322	170.6196	206.6917	121.3304
19	196.827	197.1322	170.6196	206.6917	121.3304
20	196.827	197.1322	170.6196	206.6917	121.3304
21	196.827	197.1322	170.6196	206.6917	121.3304
22	196.827	197.1322	164.3505	206.6917	121.3304
23	196.827	197.1322	164.3505	206.6917	121.3304
24	196.827	197.1322	164.3505	206.6917	121.3304
25	188.1075	197.1322	154.1458	206.6917	121.3304
26	163.5779	197.1322	153.9173	188.88	121.3304
27	153.2322	197.1322	153.9173	188.88	120.4412
28	153.2322	193.691	153.7423	188.88	120.3047
29	153.2322	193.691	146.4014	188.88	120.2426
30	153.2322	193.691	146.4014	188.88	120.1591

the proportion of discoverers was set to 20%, the scouter is 10%, and the alert threshold is set to 0.7.

To compare the performance of MSSA in the trajectory planning problem, we used two different map models as the task environment for simulation experiments. We built the three-dimensional environment and superimposed the threat area on it to make the model closer to the real environment. The area sizes of the two environments were 200×200×10 and 200×200×15 respectively. The start and end coordinates were set as $S = (0, 0, 5)$ and $E = (200, 200, 5)$. The height of the no-fly zone in the two environments was uniformly set at 10, the radius are $R_1 = [10; 8; 6; 7; 5]$ and $R_2 = [20; 15; 15]$. The center coordinates were $T_1 = [115,161; 50,150; 90,20; 175,70]$ and $T_2 = [128,161; 50,150; 60,20]$. The coordinates of the center of the mountain area were $A_1 = [48,41; 90,90; 162,96; 134,165; 30,135; 150,35]$ and $A_2 = [83,37; 149,36; 131,113; 54,118; 66,176; 131,178]$. The heights were $\mathrm{H}_1 = [7; 9;7; 6;6; 9]$ and $H_2 = [12; 8;7; 11; 6;9]$. Slopes along the X axis direction were $a_1 = [25; 25; 35; 25; 25; 20]$ and $a_2 = [25; 25; 35; 30; 25; 20]$, and slopes along the Y axis direction were $b_1 = [25; 25; 30; 20; 30; 20]$ and $b_2 = [25; 35; 30; 35; 30; 20]$. The 3-D environmental models are shown in Figure 7.

Convergence curve experiment

The trajectory simulation results of the two mission environments are shown in Figures 8 and 11, respectively. The two-dimensional trajectory planning results can be seen in Figures 9 and 12, respectively. The convergence graphs of the track cost function are in Figures 10 and 13, respectively. Figures 8 and 9 show that the five algorithms can both avoid the no-fly area from starting point to the target point in the environment. However, SSA, WOA, BOA, and TSA have large fluctuations. At the same time, the path is far away from the mountain peak and the no-fly area in the 2-D environment. This indicates that the algorithm falls into a local

TABLE 9 Statistical data of experimental results in environment 1.

Algorithm name	Optimal value	Worst value	Mean value
SSA	154.0068	221.9858	164.8511
BOA	158.9664	428.1988	193.6259
WOA	139.5953	194.8475	179.2734
TSA	171.1681	477.7604	247.9587
MSSA	130.26	223.6669	165.4328

TABLE 10 Statistical data of experimental results in environment 2.

Algorithm name	Optimal value	Worst value	Mean value
SSA	153.2322	395.4108	213.1830
BOA	193.691	411.9735	220.9971
WOA	146.4014	420.1169	189.6867
TSA	188.88	459.6827	245.5276
MSSA	120.1591	449.4203	165.6743

optimum when planning the trajectory. It is worth noting that the trajectory planned by MSSA is more stable, the fluctuation range is moderate, the trajectory is relatively stable, and it maintains a safe distance from the mountain and the threat area.

In Figure 10, the SSA converges from the seventh iteration to the 27th iteration, indicating that the original SSA algorithm fell into a local optimum during the iteration. The values of BOA and TSA decreased rapidly at the beginning of the iterations, but fell into a local optimum many times in the later stage. In contrast, MSSA found the optimal value and jumped out of the local optima many times, which resulted in an ideal cost function value, thus demonstrating that MSSA can better avoid falling into local optima and find the best path.

Figures 11, 12 show that in the second task environment, the trajectories planned by the four comparison algorithms fluctuated greatly, which confirms the poor solution quality and insufficiency of these algorithms in complex trajectory planning. The MSSA is more stable, and it can be clearly seen that the path planned by the MSSA is the shortest. Figure 13 shows that MSSA's optimal fitness values are lower than those of the comparison algorithms in the 8th iteration, and the local optimal value is still continuously removed at the end of the iteration. This shows that the trajectory cost function obtained by MSSA is smaller and the comprehensive performance index is better.

Optimization performance experiment

To more objectively compare the performance of the MSSA with that of the other four algorithms in trajectory planning, this experiment used five algorithms to conduct 30 repeated experiments in two task environments. The population size was $N = 30$, and the maximum number of iterations was $T_{\max} = 30$. The experimental results are shown in Tables 7 and 8. The statistical results of the experimental data are shown in Tables 9 and 10. The performance of each algorithm was evaluated by comparing the optimal value, worst value, average value, and standard deviation of the track cost function values obtained.

From Tables 9, 10, MSSA has obvious advantages in the performance of the UAV trajectory planning problem in the two environments. In the first environment, although MSSA is not the fastest in convergence, it did find the best value. In the second environment, MSSA was not only faster in convergence speed but also found the best value. The optimal values showed that the cost value of the MSSA for acquiring tracks was lower than that of the SSA, BOA, WOA, and TSA, illustrating that the MSSA had a stronger global search ability and higher convergence accuracy. The MSSA had a lower value than the other four algorithms on average, indicating that the MSSA had better trajectory planning quality and higher solution stability. In summary, the three improvement mechanisms proposed in this paper can effectively improve the algorithm global search ability, convergence accuracy, convergence speed and stability in terms of trajectory planning. The MSSA can balance the global search ability and local development ability of the algorithm and has excellent performance in solving complex multi-constraint combinatorial optimization problems such as UAV 3D trajectory planning. Thus, the trajectory planned by MSSA can meet the follow-up flight requirements for the UAV.

Discussion

As one of the key technologies of the UAV autonomous control system, flight path planning is a hot research area in current information science. The UAV trajectory planning problem in a 3-D environment is complicated, computationally intensive and has many local optimal solutions, which poses great challenges to the performance of optimization algorithms. Based on analysis of the path planning problem and swarm intelligence algorithm, a new sparrow algorithm was proposed. First, the algorithm adopted a reverse learning strategy based on the law of refraction, enhanced the diversity of the algorithm, and improved the optimization accuracy of the algorithm. Second, the random step size of Levy's flight boosted the local search capability of the algorithm. Lastly, the MSSA combined the fusion Cauchy mutation to increase the ability of the algorithm to jump out of local optima. We performed multiple ablation experiments in sixteen benchmark test functions with different characteristics. In the comparison of optimization performance of various improvement strategies and MSSA convergence experiments, the results indicated that Levy flight could help the population to deeply mine the global optimal values and improve the global search ability of the algorithm, proving that MSSA had greater stability and robustness than SSA.

In the comparison of convergence performance and high-dimensional performance with other algorithms, the results illustrated that MSSA had a significant competitive advantage and stability in solving complex, high-dimensional optimization problems. In addition, the Wilcoxon rank sum test and 30 CEC2014 complex functions were tested, and the results were compared with other metaheuristic algorithms and improved algorithms. The experimental results demonstrated that the improved MSSA algorithm had better stability, convergence accuracy, and optimization performance than other algorithms. Despite the significantly better performance demonstrated in this paper, the algorithm strategy could still be further improved. In subsequent work, refinements to the improved MSSA will be compared and analyzed with other advanced optimization algorithms to further improve the global search ability and local development ability of the algorithm.

Conclusion

An improved sparrow search optimization technique termed MSSA was proposed to increase the effectiveness of UAV trajectory planning in a three-dimensional environment. UAV trajectory planning simulation experiments were carried out in two different established three-dimensional geographic environments. The results showed that the path length of the obtained trajectory was considerably shorter while satisfying the constraints, further proving the feasibility and applicability of the proposed MSSA for trajectory planning in a three-dimensional environment. However, in this paper, the UAV simulation was simplified to a particle and a fixed flight speed was preset for solving the UAV path planning problem. In subsequent work, a more realistic UAV dynamics model should be established, and the flight restrictions of the UAV should be further studied. The UAV trajectory planning problem can be solved using the enhanced MSSA algorithm in subsequent work to significantly enhance the performance of complex environment trajectory planning from a variety of perspectives.

Data availability statement

The original contributions presented in the study are included in the article/Supplementary Material, further inquiries can be directed to the corresponding authors.

Author contributions

QH and LY provided the idea of the article. YY wrote the article. YY derived the formula for the trajectory planning in the article. QH and LY were responsible for checking and improving the language in the article, and QH and LY provided all of the research funding.

Funding

The research was funded by the National Natural Science Foundation of China "Research on the Evidence Chain Construction from the Analysis of the Investigation Documents (No. 62166006)", the National Natural Science Foundation of China "Rural spatial restructuring in poverty-stricken mountainous areas of Guizhou based on spatial equity: A case study of Dianqiangui Rocky Desertification Area (No. 41861038)", and Guizhou Provincial Science and Technology Projects (Guizhou Science Foundation-ZK [2021] General 335).

Conflict of interest

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

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

Abualigah, L., Yousri, D., Abd Elaziz, M., Ewees, A. A., Al-qaness, M. A. A., and Gandomi, A. H. (2021). Aquila optimizer: A novel meta-heuristic optimization algorithm. *Comput. Industrial Eng.* 157, 107250. doi:10.1016/j.cie.2021.107250

Bagherian, M., and Alos, A. (2015). 3D UAV trajectory planning using evolutionary algorithms: A comparison study. *Aeronaut. J.* 119 (1220), 1271–1285. doi:10.1017/S0001924000011246

Belge, E., Altanof, A., and Hacıoğlu, L. (2022). Metaheuristic optimization-based path planning and tracking of quadcopter for payload hold-release mission. *Electronics* 11 (8), 1208. doi:10.3390/electronics11081208

Chang, L., Shan, L., Jiang, C., and Dai, Y. (2021). Reinforcement based mobile robot path planning with improved dynamic window approach in unknown environment. *Auton. Robots* 45 (1), 51–76. doi:10.1007/s10514-020-09947-4

Derrac, J., Garcia, S., Molina, D., and Herrera, F. (2011). A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm Evol. Comput.* 1 (1), 3–18. doi:10.1016/j.swevo.2011.02.002

Dübel, S., and Schumann, H. (2017). Visualization of features in 3D terrain. *ISPRS Int. J. Geoinf.* 6 (11), 357. doi:10.3390/ijgi6110357

Fan, X., Sun, Z., Tian, E., Yin, Z., and Cao, G. (2022a). Medical image contrast enhancement based on improved sparrow search algorithm. *Int. J. Imaging Syst. Technol.* 123–23. doi:10.1002/ima.22794

Faramarzi, A., Heidarinejad, M., Stephens, B., and Mirjalili, S. (2020). Equilibrium optimizer: A novel optimization algorithm. *Knowledge-Based Syst.* 191, 105190. doi:10.1016/j.knosys.2019.105190

Hashim, F. A., Hussain, K., Houssein, E. H., Mabrouk, M. S., and Al-Atabany, W. (2021). Archimedes optimization algorithm: A new metaheuristic algorithm for solving optimization problems. *Appl. Intell. (Dordr).* 51 (3), 1531–1551. doi:10.1007/s10489-020-01893-z

Heidari, A. A., Mirjalili, S., Faris, H., Aljarah, H., Mafarja, M., and Chen, H. (2019). Harris hawks optimization: Algorithm and applications. *Future gener. Comput. Syst.* 97, 849–872. doi:10.1016/j.future.2019.02.028

Jia, Y., Zhou, S., Zeng, Q., Li, C., Chen, D., Zhang, K., et al. (2022). The UAV path coverage algorithm based on the greedy strategy and ant colony optimization. *Electronics* 11 (17), 2667. doi:10.3390/electronics11172667

Kaur, S., Awasthi, L. K., Sangal, A. L., and Dhiman, G. (2020). Tunicate swarm algorithm: A new bio-inspired based metaheuristic paradigm for global optimization. *Eng. Appl. Artif. Intell.* 90, 103541. doi:10.1016/j.engappai.2020.103541

Li, C., Zhang, N., Lai, X., Lai, X., Zhou, J., and Xu, Y. (2017). Design of a fractional-order PID controller for a pumped storage unit using a gravitational search algorithm based on the Cauchy and Gaussian mutation. *Inf. Sci.* 396, 162–181. doi:10.1016/j.ins.2017.02.026

Marini, F., and Walczak, B. (2015). Particle swarm optimization (PSO). A tutorial. *Chemom. Intelligent Laboratory Syst.* 149, 153–165. doi:10.1016/j.chemolab.2015.08.020

Mirjalili, S., and Lewis, A. (2016). The whale optimization algorithm. *Adv. Eng. Softw.* 95, 51–67. doi:10.1016/j.advengsoft.2016.01.008

Mirjalili, S., Mirjalili, S., and Lewis, A. (2014). Grey wolf optimizer. *Adv. Eng. Softw.* 69, 46–61. doi:10.1016/j.advengsoft.2013.12.007

Mohamed, N., Al-Jaroodi, I., Idries, A., and Mohammed, F. (2020). Unmanned aerial vehicles applications in future smart cities. *Technol. Forecast. Soc. Change* 153, 119293. doi:10.1016/j.techfore.2018.05.004

Momoh, J. A., Adapa, R., and El-Hawary, M. E. (1999). A review of selected optimal power ftow literature to 1993 Part I: Non linear and quadratic programming approaches. *IEEE Trans. Power Syst.* 14 (1), 96–104. doi:10.1109/59.744492

Olofsson, M., Andersson, G., and Söder, L. (1995). Linear programming based optimal power flow using second order sensitivities. *IEEE Trans. Power Syst.* 10 (3), 1691–1697. doi:10.1109/59.466472

Poli, R., Kennedy, J., and Blackwell, T. (2007). Particle swarm optimization. *Swarm Intell.* 1 (1), 33–57. doi:10.1007/s11721-007-0002-0

Price, K. V. “Differential evolution vs. the functions of the 2/sup nd/ICEO,” in Proceedings of the 1997 IEEE International Conference on Evolutionary Computation (ICEC ’97), 153–157. April 1997 Indianapolis, IN, USA doi:10.1109/ICEC.1997.592287

Qadir, Z., Ullah, F., Munawar, H. S., and Al-Turjman, F. (2015). Addressing disasters in smart cities through UAVs path planning and 5G communications: A systematic review. *Comput. Commun.* 168, 114–135. doi:10.1016/j.comcom.2021.01.003

Salgado, R., Brameller, A., and Aitchison, P. (1990). Optimal power flow solutions using the gradient projection method. Part 1. Theoretical basis. *IEE Proc. C Gener. Transm. Distrib. UK.* 137 (6), 424–428. doi:10.1049/ip-c.1990.0057

Shin, J., and Bang, H. (2020). UAV path planning under dynamic threats using an improved PSO algorithm. *Int. J. Aerosp. Eng.* 2020, 1–17. doi:10.1155/2020/8820284

Tejani, G. G., Savsani, V. J., Patel, V. K., and Mirjalili, S. (2018). Truss optimization with natural frequency bounds using improved symbiotic organisms search. *Knowl. Based. Syst.* 143, 162–178. doi:10.1016/j.knosys.2017.12.012

Thompson, S. E., and Patel, R. V. (1987). Formulation of joint trajectories for industrial robots using B-splines. *IEEE Trans. Ind. Electron.* 34 (2), 192–199. doi:10.1109/tie.1987.350954

Tinney, W. F., Member, S., and Hart, C. E. (1967). Power flow solution by Newton’s method. *IEEE Trans. Power Apparatus Syst.* PAS-86(86), 1449–1460. doi:10.1109/TPAS.1967.291823

Van Steen, M., and Leiba, B. (2018). Smart cities. *IEEE Internet Comput.* 23 (1), 7–8. doi:10.1109/MIC.2018.2887182

Videras Rodríguez, M., Melgar, S. G., Cordero, A. S., and Márquez, J. M. A. (2021). A critical review of unmanned aerial vehicles (UAVs) use in architecture and urbanism: Scientometric and bibliometric analysis. *Appl. Sci. (Basel).* 11 (21), 9966. doi:10.3390/app11219966

Wen, X., Ruan, Y., Li, Y., Xia, H., Zhang, R., Wang, C., et al. (2022). Improved genetic algorithm based 3-D deployment of UAVs. *J. Commun. Netw.* 24 (2), 223–231. doi:10.23919/JCN.2022.000014

Whitley, D. (1994). A genetic algorithm tutorial. *Stat. Comput.* 4 (2), 65–98. doi:10.1007/BF00175354

Wu, C., Fu, X., Pei, J., and Dong, Z. (2021). A novel sparrow search algorithm for the traveling salesman problem. *IEEE Access* 9, 153456–153471. doi:10.1109/ACCESS.2021.3128433

Xue, J., and Shen, B. (2020). A novel swarm intelligence optimization approach: Sparrow search algorithm. *Syst. Sci. Control Eng.* 8 (1), 22–34. doi:10.1080/21642583.2019.1708830

Zhang, D., Xian, Y., Li, J., Lei, G., and Cang, Y. “UAV path planning based on chaos ant colony algorithm[C],” in Proceedings of the International Conference on Computer Science & Mechanical Automation (IEEE), 81–85. doi:10.1109/CSMA.2015.23October 2015 Hangzhou, China .

Zhang, W., Zhang, S., Wu, F., and Wang, Y. (2021). Path planning of UAV based on improved adaptive Grey wolf optimization algorithm. *IEEE Access* 9, 89400–89411. doi:10.1109/ACCESS.2021.3090776

Zhang, X., and Duan, H. (2015). An improved constrained differential evolution algorithm for unmanned aerial vehicle global route planning. *Appl. Soft Comput.* 26, 270–284. doi:10.1016/j.asoc.2014.09.046

Zhao, Y, J., Zheng, Z., and Liu, Y. (2018). Survey on computational-intelligence-based UAV path planning. *Knowl. Based. Syst.* 158, 54–64. doi:10.1016/j.knosys.2018.05.033

Fan, Y., Fang, X., Gao, F., Zhou, X., Li, H., Jin, H., and Song, Y., (2022b). Obstacle Avoidance Path Planning for UAV Based on Improved RRT Algorithm. *Discrete Dynamics in Nature and Society*. 2022, 1, 9. doi:10.1155/2022/4544499