J. Meas. Sci. Instrum., 2025, 16(2): 245-257. http://jmsi.publish.founderss.cn/
DOI: 10.62756/jmsi.1674-8042.2025024 jmsi@nuc.edu.cn

Unmanned surface vehicles path planning with improved sparrow search algorithm

YU Hao, WANG Xin, PENG Hao*

School of Marine Engineering, Jimei University, Xiamen 361000, China

***Corresponding author:** PENG Hao (202221035095@jmu.edu.cn)

Received: February 7, 2025 **Revised:** April 18, 2025 **Accepted:** May 29, 2025

Abstract: To enable optimal navigation for unmanned surface vehicle (USV), we proposed an adaptive hybrid strategy-based sparrow search algorithm (SSA) for efficient and reliable path planning. The proposed method began by enhancing the fitness function to comprehensively account for critical path planning metrics, including path length, turning angle, and navigation safety. To improve search diversity and effectively avoid premature convergence to local optima, chaotic mapping was employed during the population initialization stage, allowing the algorithm to explore a wider solution space from the outset. A reverse inertia weight mechanism was introduced to dynamically balance exploration and exploitation across different iterations. The adaptive adjustment of the inertia weight further improved convergence efficiency and enhanced global optimization performance. In addition, a Cauchy-Gaussian hybrid update strategy was incorporated to inject randomness and variation into the search process, which helped the algorithm escape local minima and maintain a high level of solution diversity. This approach significantly enhanced the robustness and adaptability of the optimization process. Simulation experiments confirmed that the improved SSA consistently outperformed benchmark algorithms such as the original SSA, PSO, and WMR-SSA. Compared with the three algorithms, in the simulated sea area, the path lengths of the proposed algorithm are reduced by 21%, 21%, and 16%, respectively, and under the actual sea simulation conditions, the path lengths are reduced by 13%, 15%, and 11%, respectively. The results highlighted the effectiveness and practicality of the proposed method, providing an effective solution for intelligent and autonomous USV navigation in complex ocean environments.

Key words: fitness function; steering angle; chaotic mapping; inverted inertia weights; Cauchy distribution; sinusoidal chaotic mapping

0 Introduction

With the development of intelligent and information technology, unmanned surface vehicle (USV), is gradually developing towards intelligent ships. In the research related to unmanned vessels, path planning has been widely emphasized as an important class of intelligent development direction.

The development of intelligent and information technology has led to an increased focus on USV, which is evolving toward intelligent ships. Robot path planning algorithms are mainly divided into global path planning algorithms and local path planning algorithms. Currently widely used path planning algorithms are mainly based on search, sampling, or optimization path planning algorithms. Search-based path planning algorithms are mainly A* algorithm[1], D* algorithm[2], etc., which can better ensure that the optimal path (such as the shortest path or the lowest cost path) is found in known environments, but such algorithms usually suffer from high computational cost, local optima, and redundant paths, the existence of a locally optimal solution and the possibility of redundant paths or too many search nodes during the search process, which increases the computational complexity. Search process may produce redundant paths or too many search nodes, increasing the complexity of computation and other defects[3]. The main advantage of sampling-based planning algorithms is their wide applicability suitable for a variety of environments, including dynamic and static environments, high-dimensional space and so on. In addition, it also has the advantages of generating paths fast and strong exploration ability[4]. The optimization-based path planning algorithms, such as particle swarm algorithm (PSO) [5]and genetic algorithm[6], possess the advantages of strong global search ability, and they can be applied to dynamic environments and high-dimensional space. However, they still have the

disadvantages of high computational complexity and difficulty in parameter setting. Sparrow search algorithm (SSA) is a new type of swarm intelligence optimization algorithm[7]. It is inspired by the foraging and antipredator behavior of sparrows. In nature, sparrow groups search for food and avoid the threat of predators through division of labor and vigilance. SSA simulates this process by dividing sparrows into two types of roles, discoverers and followers, and introduces a detection and warning mechanism to improve the search efficiency and global optimization capability.

SSA has been widely used in the fields of path planning, data clustering, function optimization, etc. Due to its advantages of simple structure and fast convergence speed, researchers have improved and extended SSA in various ways to enhance its adaptability and performance in complex problems. Zhang et al.[8] proposed an improved SSA for mobile robots by designing a linear path strategy, which could convert the corner of the path into a smooth line in order to make the robot reach the goal faster. Liu et al.[9] proposed a sparrow population initialization strategy based on Cauchy's inverse learning, which avoided premature convergence of the algorithm. Chen et al.[10] solved the problem of redundant as well as non-smooth robot paths by combining SSA and A* algorithms with Bessel curve optimization. He et al.[11] proposed a strategy that introduced a nonlinear dynamic weighting factor, which reduced the algorithm's dependence on the location of the producer and balanced its global and local exploration capabilities. Liu et al. [12] designed a comprehensive evaluation metric to evaluate the differences in the paths generated by different path planning algorithms.

There are still few studies on its application of SSA in the USV[13]. Given the particularity of the movement of USV[14-16], as well as the physical limitations of steering angles and the number of turns during navigation, the path planning work has to take into account multiple performance indicators such as ship safety, which poses a great challenge. At present, there are few studies on path planning that considers both the optimal path and the particularity of the movement of the USV. At the same time, traditional optimization algorithms are prone to local optimal solutions, resulting in poor algorithm effects. This paper studied the relevant algorithms for the above two problems, and proposed an improved SSA.

The main contributions of this paper are as follows.

1) The fitness function was redesigned to consider path length, maximum turning angle, and planning time, addressing the limitations of traditional algorithms that neglected physical constraints and safety. Through this multi-objective trade-off, the generated path was more in line with the stability and controllability requirements of actual maritime operations, improving the reliability of unmanned ships in complex environments.

2) The use of sinusoidal chaotic mapping to initialize the population effectively improved the spatial coverage rate and solved the problem of uneven initial population distribution and easy to fall into local optimality in traditional SSA. At the same time, the adaptive reverse inertia weight dynamically adjusted the search intensity according to the search stage, realizing the strategy of extensive exploration in the early stage and refinement in the later stage, improving the convergence speed, search quality, and the stability and efficiency of the algorithm.

3) By combining the large-scale perturbation of Cauchy distribution with the small-scale fine-tuning of Gaussian distribution, the jumpiness and accuracy of the path search process were enhanced, overcoming the shortcomings of traditional optimization algorithms, such as fast local convergence and easy stagnation. This strategy effectively improved the robustness and global optimization capabilities of the algorithm in multi-peak and multi-constraint problems, and enhanced the adaptability and versatility of unmanned ships in planning paths in complex waters.

1 SSA

1.1 Fundamentals

The SSA is a heuristic optimization method inspired by the foraging behavior of sparrows in nature. It models a population comprising three types of individuals: producers, followers, and scouters. The algorithm leverages the cooperative and competitive dynamics among these roles to explore the solution space. Through information sharing and group interactions, SSA iteratively searches for the global optimum.

Producers are responsible for finding food sources, and they are the leader of the group. They are the most adaptable and the best individuals in the group. Followers follow the producer and rely on the food sources found by the producer for foraging. They are less well adapted, but can obtain a better solution by following the producer. Scouters have poor adaptations and are responsible for warning other individuals in the group of danger or predation. They will always be alert and monitor the safety of the group.

The core of the SSA lies in optimizing the problem solution by simulating the sparrow's position update process during its search for food. Individual sparrows

collaborate with each other through the roles of producer, follower, and scouters to perform global and local searches. The main idea is as follows.

The first step involves updating the position of each sparrow in the algorithm. Each sparrow is represented by a position (a point in the solution space) that is iteratively updated. Depending on the type of individual (producer, follower, or scouter), the update process differs.

The producer is responsible for exploring new solutions in the search space, and its updates mainly consider its own position and the optimal solution for the current population. The updating method of the producer usually uses a combination of inertia weights and random perturbations. Its update formula is

$$p_i(t+1)=\omega p_i(t)+\alpha Best_p(t), \tag{1}$$

where ω is the inertia weight, which controls the influence of the current position on the update, and it is set to 0.5; a is the learning factor, which controls the influence of the optimal position on the current position, and it is set to 0.5. $Best_p$ is the global optimal or local optimal position; p is position.

A follower updates its position by following a producer. The way the follower updates will be based on the producer's position combined with some random factors, and the update formula is

$$p_i(t+1)=p_i(t)+\lambda(Best_p-p_i(t))+\delta, \tag{2}$$

where p is the position and λ is the adjustment factor, which controls the distance between the follower and the producer; δ is a random perturbation term, which can enhance the algorithm search capability by adding some randomness.

Scouters are responsible for monitoring their surroundings and actively avoiding dangers when they are detected. When a scouter updates its position, it usually relies on the worst-fit position to update in order to avoid falling into a local optimum. The update formula for scouters is

$$p_i(t+1)=Best_p+\beta_i(t)-\gamma(p_i(t)-Worst_p(t)), \tag{3}$$

where β and γ are constant coefficients that control the relationship between the vigilante and the optimal and worst positions, respectively.

1.2 Fitness function

The fitness of each sparrow indicates the quality of its solution. Usually, the fitness is calculated by the objective function[17,18]. Let the objective function of the problem be $f(x)$, then the fitness of the sparrow f_i is

$$f_i=f(p_i), \tag{4}$$

where $f(p_i)$ is the value of the objective function corresponding to the sparrow position.

To enhance search effectiveness, the SSA maintains a balance between global exploration and local exploitation through the cooperative and competitive interactions among different individual roles during iterations. This coordinated role-based mechanism enables the SSA to efficiently conduct both global and local searches.

SSA usually sets a maximum number of iterations T_{max} or a fitness threshold as a stopping condition. The algorithm will keep updating the individual positions during the iteration process until the stopping condition is satisfied.

SSA is a natural heuristic optimization algorithm based on the foraging behavior of sparrow groups. By modeling the roles of producers, followers, and scouters in a sparrow population, the algorithm is able to perform global search and local exploitation. Its core mathematical principles include mechanisms such as individual position updating, fitness evaluation, and balance between global and local search, which effectively enhances the optimization ability of the algorithm and avoids the shortcomings of traditional algorithms that tend to fall into local optimum.

2 Path planning based on improved SSA

In order to solve the multi-objective in the fitness function, the global search capability and optimization efficiency of the SSA are improved by optimizing the fitness function, introducing chaotic mapping, adaptive inverse inertia weights, and Cauchy-Gaussian hybrid updating strategy.

2.1 Adaptation function improvement

In order to obtain the optimal navigation path that takes into account the safety, physical constraints, and energy efficiency of USV, this paper proposes an improved SSA with respect to the fitness function, the initial population generation, the position updating strategy, and the optimization mechanism of local search.

In actual navigation, USV usually do not use a large steering angle. Once a large steering angle is used, it may lead to behaviors that affect the safety of USV such as sideways tilting. Meanwhile, it is also very important to study the path length, which is related to the working efficiency and energy consumption of USV. Therefore, the fitness function is redefined by incorporating the path length and turning angle generated by the algorithm, that is

$$fitness=\sqrt[3]{(x_i-x_0)^2+(y_i-y_0)^2}+(\theta_{max})^3+T^3, \tag{5}$$

where x_i denotes the x-coordinate of the end point i; x_0 denotes the position of the start point; y_i denotes the y-coordinate of the end point i; y_0 denotes the y-coordinate of the start point; $\theta_{\max}$ denotes the maximum steering angle; T denotes the total time spent in planning.

The path length is a fundamental metric for evaluating route efficiency. A shorter path implies lower energy consumption and reduces task completion time. Moreover, path length serves as a foundation for subsequent path-tracking control, directly influencing system accuracy, and response efficiency.

The steering angle indicates the smoothness and feasibility of the planned path. Due to the effects of inertia, water currents, and hull structure, USV is constrained in their maneuverability. Frequent or sharp turns can lead to unstable navigation and increased operational risk. Therefore, constraining the steering angle contributes to smoother trajectories, enhances dynamic feasibility, and improves control robustness.

The navigation time is a key indicator of task execution efficiency and energy management. In scenarios involving multi-USV coordination or energy-limited missions, time becomes a critical priority. To emphasize this factor, a cubic time term is incorporated into the fitness function, thereby increasing its weight and ensuring the optimized solution not only meets safety and controllability standards but also improves overall task efficiency.

The coordinated optimization of the three factors not only considers the geometric efficiency of the path (i.e., the shortest route) but also ensures navigation stability (the minimal turning) and execution efficiency (the shortest time) in dynamic environments. This approach brings the planning results closer to the practical requirements of real USV system operations. The design of the fitness function thus provides a robust evaluation framework for subsequent path optimization and control.

2.2 Chaotic mapping

Chaotic mapping initialization is a method of generating an initial population using the nonlinear properties of chaotic systems, which is often used in optimization algorithms to improve the diversity of the search and avoid the algorithm from falling into a local optimal solution. Chaotic system is highly sensitive and non-periodic, and can generate numerical sequences with good distribution, so it can effectively guide the search process for global exploration.

A chaotic system is described by a set of nonlinear dynamic equations, which usually exhibits sensitive initial condition dependence, non-periodicity, and widely distributed properties. In optimization problems, the basic idea of chaotic mapping is to use the properties of chaotic systems to generate an initial population which is widely distributed in the search space, thus improving the global search capability.

The chaotic system can be represented by

$$x_{n+1}=f(x_n),\ 0\leqslant x_n\leqslant 1, \tag{6}$$

$$x_{n+1}=\mathrm{mod}\left(z_i+a-\frac{b}{2\pi}\sin(2\pi z_i)-\frac{b}{2\pi}\cos(2\pi z_i)+r,1\right), \tag{7}$$

where x_n denotes the state of the system at the nth iteration; and $f(x_n)$ is the chaotic mapping function, which plays a decisive role in the evolution of the system; mod is the remainder function; a and b are mapping parameters. After multiple simulation experiments, this paper takes the values of $a=0.2$, $b=1.2$, and r is a random number between $(0,1)$. The use of the improved chaotic circle mapping function enhances the quality of the optimal individuals in the initial group and the population diversity, realizes the exploration of more high-quality search areas, and improves the convergence performance of the algorithm.

In the sparrow search optimization algorithm, chaotic mapping initializes the sparrow positions and keeps it within the boundary of the search space through normalization.

The primary advantage of sinusoidal chaotic mapping in path planning lies in its ability to generate a sequence that is uniformly distributed within the interval $(0,1)$, which facilitates a more comprehensive exploration of the initial solution space and enhances both population diversity and the global search capability of the algorithm. Additionally, the simplicities of its functional structure, high computational efficiency, and strong numerical stability make it well-suited to the large-scale iterative calculations required in USV path planning. Furthermore, the non-periodicity and excellent ergodicity of sinusoidal mapping confer superior adaptability and convergence performance, particularly when addressing complex multi-constraint and multi-peak path planning problems. Therefore, the selection of sinusoidal mapping for chaos initialization in this study is based on a thorough evaluation of its distribution characteristics, engineering feasibility and optimization performance.

Functional steps for initialization of chaotic mapping are as follows. First, the position of the first sparrow is initialized using a randomly generated value $x_0=\mathrm{random}(0,1)$. Secondly, each subsequent sparrow position is generated by the chaotic mapping computation. Assuming that the current sparrow's position is x_{i-1}, the

position x_i is updated by

$$x_{i,j} = \sin\left(\frac{a\pi}{x_{i-1,j}}\right), \tag{8}$$

where a is a random number and $x_{i,j}$ is the positions of the previous sparrow in the j dimension. This formula produces chaotic behavior by varying the trigonometric functions, making the generated values more widely distributed in the search space.

Finally, through several iterations of computation, the position of each sparrow $x_{i,j}$ is generated in the given range of dimensions, and it is ensured that the value of each dimension is between [0, 1]. Eventually, the initial position of the entire sparrow population is represented by a matrix, where each row represents the position of one sparrow.

Chaotic mapping offers several advantages in optimization, such as ensuring a uniform initial population distribution and improving global search efficiency. Its nonlinear dynamics also enhance sensitivity and complexity, helping the algorithm escape local optima. Additionally, due to extreme sensitivity to initial conditions, chaotic mapping generates diverse initial values in each run, overcoming the limitations of traditional random initialization. Finally, chaotic systems evolve continuously during iterations, enhancing adaptability and helping prevent premature convergence by introducing diverse solutions throughout the optimization process.

Ultimately, the locations of all sparrows are initialized by this chaotic mapping, generating a widely distributed initial population that provides diverse starting points for the subsequent optimization process.

Chaotic mapping initialization generates initial populations by exploiting the nonlinear properties of chaotic systems, which provides a wider and more uniformly distributed initial solution than traditional random initialization methods, thus improving the efficiency and quality of global search. It helps to avoid the local optimal solution problem and is particularly suitable for complex optimization problems.

2.3 Adaptive inverse inertia weights

Adaptive inverse inertia weighting is a method that dynamically adjusts the inertia weights during the individual update process in the sparrow search optimization algorithm. Inertia weights[19-21] are the degree to which an individual's current position affects velocity when updating its position. The purpose of introducing inverse inertia weights is to enable the algorithm to explore extensively in the search space at the beginning by balancing the ability to explore and exploit, while concentrating on existing quality solutions at a later stage. Adaptive tuning of the inertia weights enhances the convergence and global search capability of the algorithm.

Inertia weight is a parameter that controls the rate of particle updating, and it affects how much an individual relies on historical information in the optimization process. Larger inertia weights enhance exploration, and smaller inertia weights enhance exploitation. Reverse inertia weighting, on the other hand, refers to the use of an opposite approach to conventional inertia weighting when calculating inertia weights, usually achieved by gradually reducing the inertia weights from the maximum value to the minimum value.

In the strategy of adaptive inverse inertia weights, the inertia weights are not fixed, but dynamically change according to the individual's fitness and the number of iterations. Specifically, the inertia weights are adjusted according to the distance between the individual and the optimal solution, as well as the current stage of optimization. In this way, the algorithm is able to explore more globally in the early stages and focus more on local search and fine-grained optimization in the later stages.

In the code, the reverse inertia weights are calculated as follows.

Step 1 Calculate the inertia weights. The inertia weights are normalized by a sinusoidal function that makes the weights vary in the interval [0, 1]. Given the difference between individual fitness and optimal fitness, the difference can be utilized to dynamically adjust the inertia weights.

Assuming that the fitness of an individual is f_i, the optimal fitness is f_{best}, the worst fitness is f_{worst}. The difference in fitness is

$$temp_{x_i} = \frac{f_i - f_{\text{best}}}{f_{\text{worst}} - f_{\text{best}}}, \tag{9}$$

where f_{best} is the optimal fitness of all individuals in the current iteration; f_{worst} is the worst fitness; a $temp_{x_i}$ denotes the normalized fitness difference.

Step 2 Calculate inertia weights. Based on the normalized fitness difference, the inertia weights are adjusted by a sinusoidal function, which makes the inertia weights changeable throughout the search process.

The formula for calculating the inertia weights is

$$\omega_i = \omega_{\min} + (\omega_{\max} - \omega_{\min}) \times \frac{\sin\left(\pi \cdot temp_{x_i} + \frac{\pi}{2}\right) + 1}{2}, \tag{10}$$

where $\omega_{\min}$ is the minimum value of inertia weights and $\omega_{\max}$ is the maximum value of inertia weights. According

to experience, we set the maximum value to 0.8 and the minimum value to 0.2[22]. $temp_{x_i}$ is the normalized fitness difference, which is used to adjust according to the relative difference between the current fitness and the optimal fitness.

Fig. 1 shows the iterative curves of three commonly used inertia weight ranges. The curve within the range of 0.2 — 0.8 converges the fastest. Therefore, this range was selected as the inertia weight range.

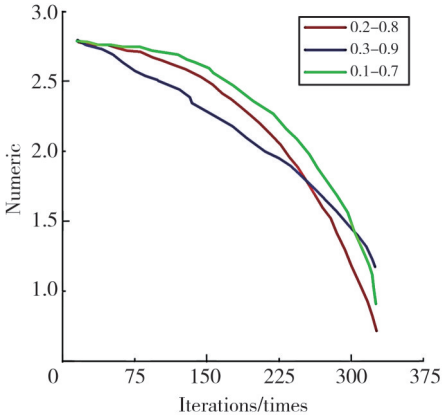


Fig. 1 Iterative curves of three common inertia weight ranges

By varying the sinusoidal function, the inertia weights vary smoothly between [0, 1], allowing individuals to explore globally (larger inertia weights) and exploit locally (smaller inertia weights) during the search process.

Step 3 Reverse inertia weights are usually realized through a reverse adjustment of the inertia weights, implying a reverse impact through a comparison of the inertia weights (e.g., the difference with the maximum inertia weight).

The reverse inertia weights are calculated by

$$\omega_{\text{inverted}} = \omega_{\max} + \omega_{\min} - \omega_i. \tag{11}$$

The reverse inertia weight, by adjusting the inertia weight, increases the algorithm's focus on the optimal solution in the later stage, enhances the development capability, and avoids excessive random exploration in the later stage of optimization. Adaptive inverse inertia weights can dynamically adjust the ratio of global exploration to local exploitation during the optimization process. Larger inertia weights at the initial stage help to conduct extensive global search and avoid falling into local optimal solutions, while smaller inertia weights at the later stage promote fine search and accelerate the convergence speed. Different from the traditional fixed inertia weights, the inertia weights adaptively change according to the fitness and optimization stage, and can be automatically adjusted according to the quality of the current solution and the progress of the search, thus enhancing the adaptability of the algorithm. In addition, by introducing reverse inertia weights, the gradual reduction of inertia weights in the later stages of optimization allows individuals to focus more on the development of high-quality solutions, which in turn accelerates the convergence process. Meanwhile, larger inertia weights ensure that the algorithm is able to conduct extensive exploration in the early stage. While smaller inertia weights help the algorithm to focus on the neighborhood of the current optimal solution, which effectively prevents premature convergence and falling into local optimal solutions.

2.4 Cauchy-Gaussian mixture update strategy

The Cauchy-Gaussian hybrid update strategy is an update strategy that combines the Cauchy distribution and the Gaussian distribution (normal distribution). Its main purpose is to enhance the stochasticity and diversity of the search process by mixing the characteristics of these two distributions to help the optimization algorithm jump in the solution space, thus avoiding falling into the local optimal solution and improving the global search capability.

Cauchy-Gaussian distributions have their advantages in stochastic optimization. Gaussian distribution is centralized and suitable for fine search in the vicinity of the current solution. Its random perturbations are small and can help the optimization algorithm to search locally. The Cauchy distribution has a heavy-tailed nature and helps in escaping from the local optimum by allowing a larger range of jumps in the search space as compared to the Gaussian distribution. The random perturbations are larger in the Cauchy distribution, and they can help in exploring the entire solution space. Combining these two distributions allows for larger random perturbations in the global search, while finely searching the region around the current solution through the Gaussian distribution at convergence, thus effectively balancing the ability to explore and exploit.

The Cauchy-Gaussian hybrid update strategy is

$$p[p_i;:] = \omega_i \cdot p[p_i;:] + q \cdot cauchy_g auss, \tag{12}$$

where q is a standard normally distributed random number generated with values following $N(0,1)$. Cauchy-Gaussian distributions is a perturbation term generated through a mixture of Cauchy and Gaussian distributions. Specifically, the mixed perturbation term is generated by

$$cauchy_g auss = 1 + R_2 \cdot \text{Cauchy}(0,1) + (1 - R_2) \cdot N(0,1), \tag{13}$$

where R_2 is a random number in the range [0, 1] that controls the ratio of the Cauchy distribution to the Gaussian distribution. If R_2 is close to 1, more Cauchy

distributions are used; if R_2 is close to 0, more Gaussian distributions are used. Cauchy(0, 1) denotes a standard Cauchy-distributed random variable with heavy-tailed properties. $-N(0, 1)$ denotes a standard normally distributed random variable.

The specific steps of the hybrid strategy are as follows.

Step 1 A hybrid perturbation term is computed for each individual producer. The perturbation term combines the properties of the Cauchy and Gaussian distributions. The Cauchy distribution produces larger perturbations and then explores farther solution space, while the Gaussian distribution is better suited for localized searches. R_2 controls the mixing ratio of the Cauchy-Gaussian distributions. By introducing R_2, the algorithm has the flexibility to adjust the balance between exploration and exploitation at different stages.

Step 2 For each producer individual, the position is updated by

$$p[p_i,:]=\omega_i \cdot p[p_i,:]+ q \cdot cauchy_g auss, \tag{14}$$

where ω_i is the inertia weight of the current individual; q is a random perturbation with a standard normal distribution. This formula combines the inertia weights and the mixture of perturbation terms that affect the movement of an individual in the search space.

Step 3 In order to prevent individuals from jumping out of the search space, the code performs boundary processing on individual positions. This step ensures that the position of each individual is within the given search space boundary. The principle is

$$p[p_i:]= \mathrm{clip}(p[p_i,:], low_{\mathrm{boundary}}, up_{\mathrm{boundary}}). \tag{15}$$

Step 4 After updating the position, the adaptation value of the new position is calculated by the adaptation function, and the optimal adaptation and position are updated.

The heavy-tailed nature of the Cauchy distribution allows the strategy to perform a wide range of random perturbations during the search process, which enhances the global search capability and prevents the algorithm from falling into a local optimum. Meanwhile, the Gaussian distribution provides smaller random perturbations, enabling individuals to perform fine local search within the neighborhood of the current solution, which helps the algorithm to converge to the global optimum solution more quickly and accurately. In addition, the ratio of the Cauchy distribution to the Gaussian distribution can be dynamically adjusted by the random number R_2 to achieve a balance between global exploration and local exploitation, and to adapt to the optimization needs at different stages. The introduction of the Cauchy distribution also prevents the algorithm from converging prematurely in the early stage of the search, which further improves the global exploration capability and enhances the overall robustness of the algorithm.

2.5 Overall implementation process

The SSA significantly improves the optimization efficiency and search capability through several innovative points. First, the role assignment mechanism balances exploration and exploitation by dividing individuals into producers, followers, and scouters, which assumes the roles of global exploration, local exploitation, and local optimal avoidance, respectively. Second, the dynamic role reassignment based on fitness ensures that the algorithm always focuses on solutions with higher potential. Diverse update strategies enhance flexibility, effectively improving the ability of global and local search. Finally, the adaptive inertia weighting and inverse inertia weighting mechanisms are able to adjust the ratio of exploration to exploitation according to the search phase, accelerating convergence and avoiding the trapping of locally optimal solutions. These innovative points make SSA have higher efficiency and stronger global search ability in complex optimization problems.

3 Performance test

3.1 Benchmark function test experiment

To verify the effectiveness of the algorithms, four benchmark functions are selected for performance testing: two single-peak functions and two multi-peak functions. The results are shown as Table 1.

In the benchmark function testing, two single-peak and two multi-peak functions are employed to evaluate and compare the optimality of the algorithms. The problem dimension is set to 600, with respective search ranges of $[-100, 100]$, $[-100, 100]$, $[-5.12, 5.12]$ and $[-600, 600]$.

Experiments were conducted using PyCharm 2022.3 on a computer running Windows 11, equipped with an Intel i7-6700 processor (3.40 GHz) and 32 GB of RAM. Table 1 presents the results from 40 independent runs of each algorithm on the benchmark functions.

The four benchmark functions are illustrated in Figs.2—5. These classical test functions are widely used to evaluate the performance of optimization algorithms, and they are employed here as benchmarks.

Table 1 Benchmark function tests

Typology	Function name	Dimension	Population size	Scope of exploration	Fitness value
Single peak	Weighted sum of squares function	600	100	[−100,100]	7.575 6e-3
	Rosenbrock's valley function	600	100	[−100,100]	5.682 1e-3
Multi-peak	Combined function of weighted absolute value sum and product	600	100	[−5.12,5.12]	1.030 3e-1
	Griewank function variants	600	100	[−600,600]	5.002 0e-3

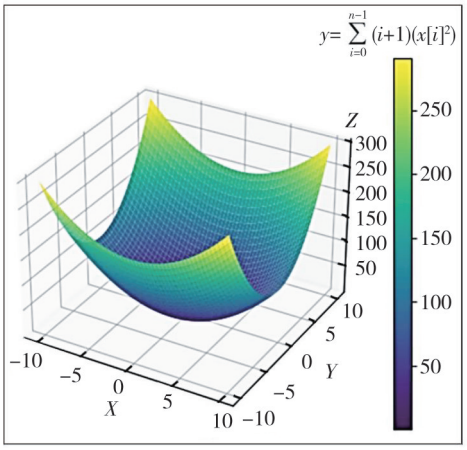


Fig. 2 Function I: Weighted sum of squares function test results

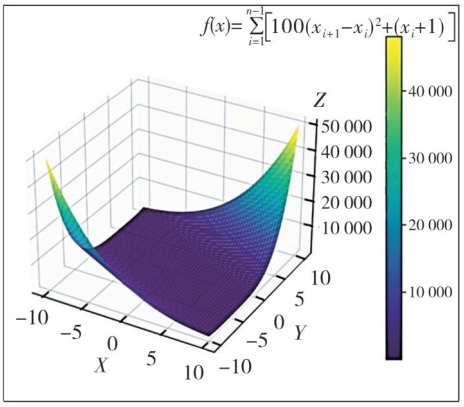


Fig. 3 Function II: Rosenbrock's valley function test results

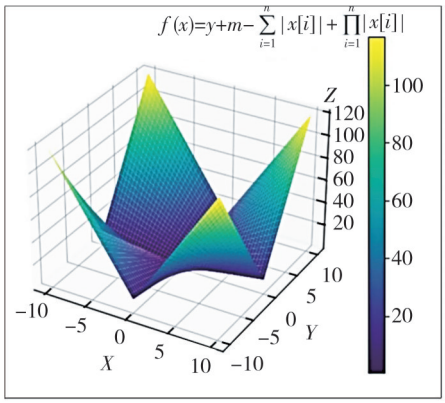


Fig. 4 Function III: Combined function of weighted absolute value sum and product test results

Figs.6—8 present the performance comparison between the improved SSA and other optimization algorithms across the four benchmark functions. Within 300 iterations, the improved SSA consistently demonstrates faster convergence and superior optimization performance.

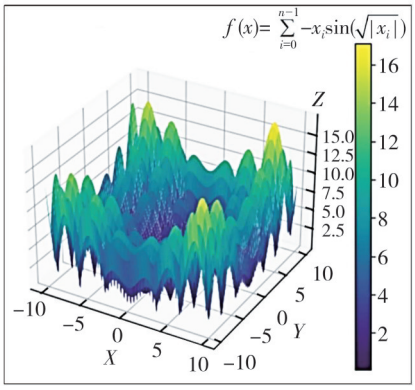


Fig. 5 Function IV: Griewank function variants test results

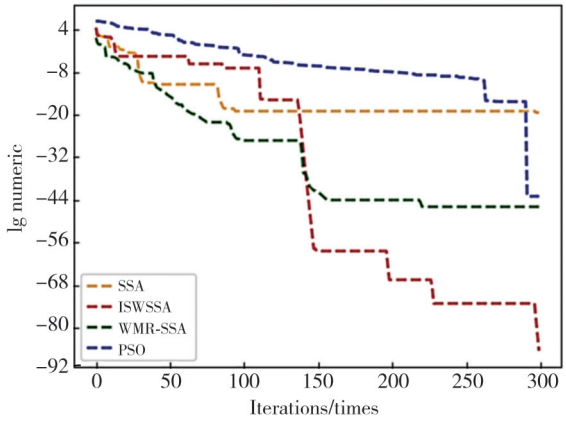


Fig. 6 Convergence curves of function I

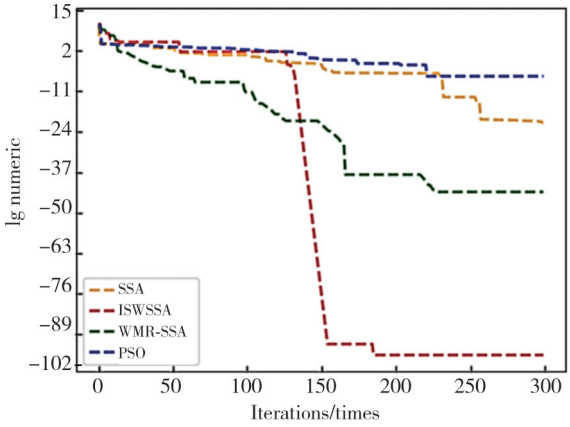


Fig. 7 Convergence curves of function II

The optimal fitness values are presented in Table 2. It compares the performance of four algorithms (SSA, improved SSA, WMR-SSA, and PSO) across four different fitness functions

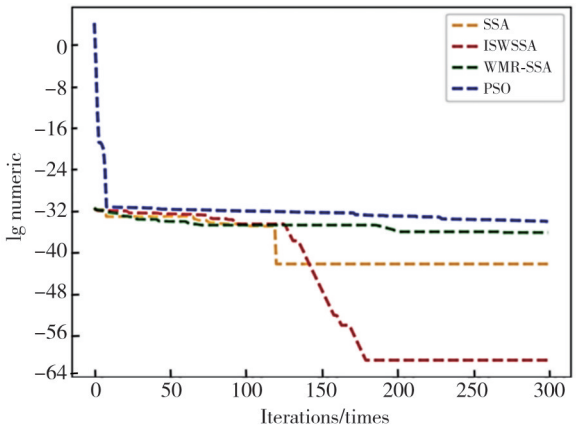


Fig. 8 Convergence curves for function IV

Table 2 Optimal fitness of each algorithm

Function	SSA	Improved SSA	WMR-SSA	PSO
Weighted sum of squares function	4.741e-20	1.216e-82	0.003	1.191e-43
Rosenbrock's valley function	9.303e-22	2.848e-75	0.021	5.974e-06
Combined function	0.019	4.413e-49	1.879e-10	4.701e-07
Griewank function	0.099	5.161e-29	1.248e-06	3.927e-07

3.2 Computational complexity analysis

The time complexity of the SSA is compared with that of the improved SSA. The time complexity of SSA is $O(d+f(d))$. Among them, d is the dimension and $f(d)$ is the time to solve the objective function. The time used for population initialization is δ_1, and the time used to generate each dimension of the chaotic map is δ_2.

The discoverer position update strategy adopts the introduction of nonlinear normal distribution coefficients, and the time complexity is $T_2=O(r_1N(\delta_3+\delta_4)d)$, where δ_2 is the proportion of the population occupied by discoverers in the algorithm. There is a random number in the position update formula, and the time for random generation is δ_3, and the time spent on position update in each dimension is δ_4.

The Cauchy-Gaussian distribution strategy is introduced to help the algorithm escape from the local optimum. The strategy is generally not used for the entire population in every round, but only for elite individuals trapped in the local optimum (The number is much smaller than the population size N). The average computational cost of this strategy can be expressed as a ratio $\rho\in(0,1)$ to the number of applied individuals.

The time complexity of the Cauchy-Gaussian strategy is $T_3=O(\rho N\cdot d)$, considering $\rho\ll 1$. This term is of the same order of magnitude as the main update strategy, but does not dominate the total complexity. Therefore, the total time complexity of the improved SSA is

$$T_{\text{total}}=O(Nd+iter_{\max}\cdot Nd+iter_{\max}\cdot\rho Nd)\approx O(iter_{\max}\cdot Nd). \tag{16}$$

And the time complexity used by SSA and improved SSA is basically the same. On the basis of maintaining the original SSA linear time complexity, the ability to jump out of the local optimal solution is improved.

4 Simulation experiments

4.1 Environment modeling

The working environment of a surface unmanned vessel can be regarded as a two-dimensional space. The raster method[23,24] was used to model the environment, where the size of the area of each raster depended on the size of the USV. In order to ensure the safety of the planned route, the obstacles are inflated in this paper with an inflation radius of 0.5. Fig. 9 illustrates the map of the simulated environment, while Fig. 10 presents the corresponding map of the actual maritime area. The parameters of improved SSA are shown in Table 3.

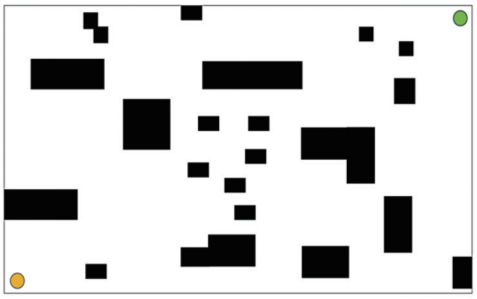


Fig. 9 Simulated sea map environment

Fig . 10 Kurgan sea chart

Table 3 Parameters and corresponding values related to improved SSA

Parameter	Numerical value
Maximum number of iterations	300
Population size	30
Safety threshold	0.9
Number of discoverers	5
Number of observers	6

Fig. 11 depicts the simulated environment map corresponding to the actual sea area shown in Fig. 10.

Fig. 11 Actual sea area map environment

4.2 Evaluation functions

In order to evaluate the performance index of the algorithm, the comprehensive performance index was analyzed here using hierarchical analysis to establish the evaluation function. Path length (the maximum steering angle of the pat) and path planning time are selected as evaluation criteria, and certain weights are assigned to calculate the performance of the algorithm, that is

$$s = \alpha_1 w_1 + \alpha_2 w_2 + \alpha_3 w_3, \tag{18}$$

where α_i, $i = 1,2,3$, represent the corresponding weights of the three indicators which are designed to be 0.4, 0.3, and 0.3, respectively; w_1, w_2, w_3 represent the path length, the maximum steering angle, and the path planning time, respectively. The calculation uses the maximum value minus the deviation of the current path, normalized to ensure that strong performance in a single aspect does not disproportionately influence the overall score.

4.3 Simulation experiments in simulated sea area

Fig. 12 shows the path planning results of the four algorithms in the first map, respectively, while Fig. 13 presents the corresponding convergence curves.

Table 4 shows the simulation settings for the PSO algorithm. In the simulated sea area, as shown in Table 5, the improved SSA achieves not only the shortest path length but also the smallest maximum steering angle[25], at just 0.81°. Since larger steering angles deviate from the normal motion of a USV, this result indicates better conformity to real-world movement constraints. Furthermore, the improved SSA exhibits shorter path planning times than both the SSA and PSO algorithms, and it is comparable to the WMR-SSA algorithm. Faster planning time enhances the algorithm's applicability to real-time USV navigation. Finally, based on the evaluation function, the improved SSA achieves the highest score of 1.0, while the PSO algorithm receives the lowest. In terms of overall performance, the proposed method outperforms the SSA and WMR-SSA algorithms, respectively.

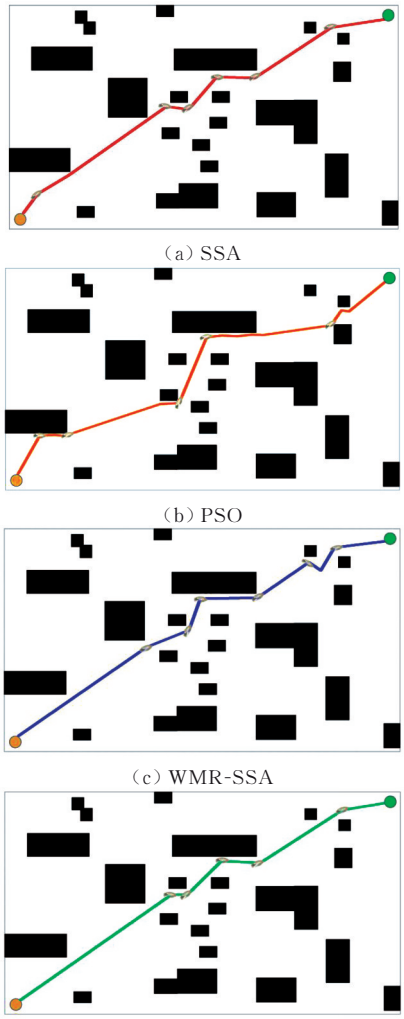


Fig. 12 Path planning results in simulated sea area

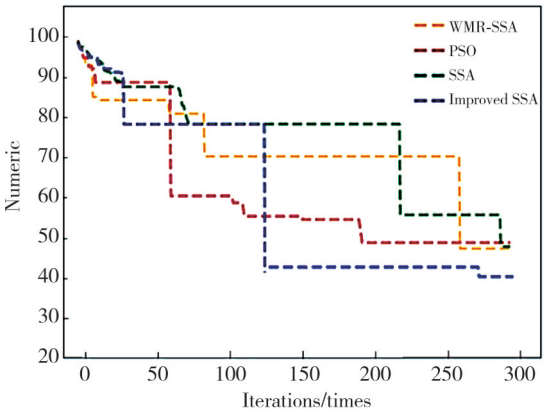


Fig. 13 Convergence plot of the shortest path in simulated sea area

Table 4 Parameters and corresponding values related to PSO algorithm

Parameter	Numerical value
Maximum number of iterations	300
Number of groups	30
Learning factor	1.5
Inertial weighting	0.4−0.8

Table 5 Comparison of experimental data of simulated sea environment

Simulated marine environment	Path length/m	Maximum steering angle/rad	Planning time/s	Composite indicators
SSA	49.72	1.12	0.31	0.37
PSO	49.87	1.49	0.37	0.00
WMR-SSA	47.98	0.97	0.28	0.62
Improved SSA	41.22	0.81	0.28	1.00

4.4 Simulation experiments in actual sea area

Compared with the simulated sea area, the actual sea area is more open and has fewer dispersed obstacles, which requires readjustment of the algorithm-related parameters. Table 6 shows the parameter settings of SSA, WMR-SSA, and improved SSA. Table 7 shows the parameter settings of PSO[26].

Table 6 Parameters and values related to improved SSA

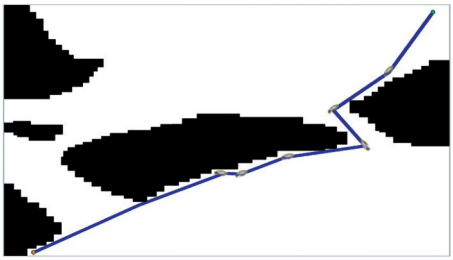
Parameter	Numerical value
Maximum number of iterations	300
Population size	40
Safety threshold	0.9
Number of discoverers	6
Number of observers	6

Table 7 Parameters and values related to PSO algorithm

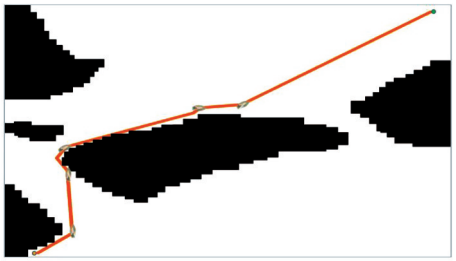
Parameters	Numerical value
Maximum number of iterations	300
Number of groups	40
Learning factor	1.5
Inertial weighting	0.3—0.8

Fig. 14 illustrates the simulated path planning results of the four algorithms in the second map, which represents the actual maritime environment. Fig. 15 presents the corresponding convergence curves.

(a) SSA



(b) WMR-SSA



(c) PSO

(d) Improved SSA

Fig. 14 Path planning for actual sea area

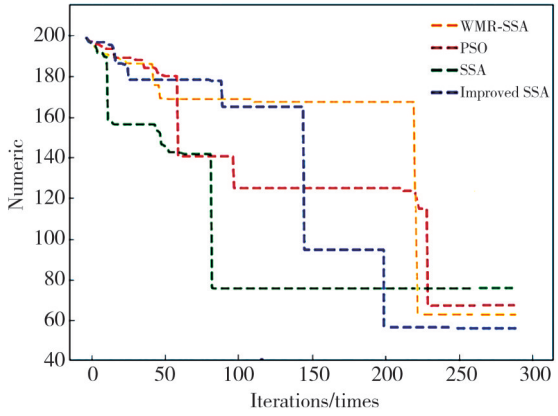


Fig. 15 Convergence diagram of the shortest path in actual sea area

Table 8 shows the comparison of experimental results. It shows that the improved SSA outperforms the comparison algorithms in terms of path length, the maximum steering angle, and planning time.

Compared to the SSA algorithm, PSO algorithm, and WMR-SSA algorithm, in the simulated sea area, the path lengths of the improved SSA are reduced by 21%, 21%, and 16%, respectively, and under the actual sea simulation conditions, the path lengths are reduced by 13%, 15%, and 11%, respectively. And its comprehensive scores are higher than the other algorithms.

Table 8 Comparison of experimental data in actual sea environment

Simulated marine environment	Path length/m	Maximum steering angle/rad	Planning time/s	Composite indicator
SSA	57.24	1.07	0.57	0.15
PSO	58.27	1.19	0.41	0.22
WMR-SSA	55.91	0.86	0.38	0.65
Improved SSA	50.46	0.82	0.35	1.00

5 Conclusions

In order to obtain high-performance USV paths, an improved SSA based on adaptive hybrid strategy was proposed. By optimizing the fitness function, path length and steering angle, as well as the safety and energy efficiency of USV, were taken into account so as to better meet the actual navigation requirements. The introduction of chaotic mapping initialization, inverse inertia weights, and Cauchy-Gaussian hybrid update strategy has enhanced both global search capability and convergence speed. This approach effectively avoided local optima and improved the algorithm's applicability in complex environments.

The improved SSA demonstrated better path planning effects than the traditional algorithm in several test scenarios, and exhibited strong adaptability and robust performance. The algorithm was able to balance global exploration and local optimization, which improved computational efficiency while ensuring path planning accuracy.

Despite promising results, this study has some limitations. The method needs further validation in highly dynamic or obstacle-rich 2D environments. Its scalability to 3D scenarios or coordinated multi-USV operations also requires further investigation to broaden its applicability.

The improved SSA has a good application prospect, especially in autonomous navigation and mission execution in complex water environment. To further support operation in complex dynamic environments, future research may incorporate motion prediction techniques such as Kalman filtering or learning-based models to anticipate obstacle trajectories and improve real-time responsiveness. For 3D scenarios, the algorithm will be extended by incorporating altitude or depth into the state space and utilizing 3D spatial representations to ensure accurate and safe navigation in volumetric environments.

Acknowledgement

This work was supported by Shandong Provincial Department of Science and Technology Project (No. 2022C01246); National Undergraduate Innovation Training Project (Nos.202410390028, 202310390026), Fujian Provincial Undergraduate Innovation Training Project (No. 202410390093), and Jimei University Innovation Training Project (Nos.2024xj224, 2023xj179).

Declaration of conflicting interests

All the authors have no conflict of interests related to this publication.

References

[1] TANG G, TANG C, CLARAMUNT C, et al. Geometric a-star algorithm: an improved a-star algorithm for AGV path planning in a port environment. IEEE Access, 2021, 9: 59196-59210.

[2] CAI X, WU L, ZHAO T, et al. Dynamic adaptive multi-objective optimization algorithm based on type detection. Information Sciences, 2024, 654: 119867.

[3] HE Z, LIU C, CHU X, et al. Dynamic anti-collision a-star algorithm for multi-ship encounter situations. Applied Ocean Research, 2022, 118: 102995.

[4] WU Z, MENG Z, ZHAO W, et al. Fast-RRT: A RRT-based optimal path finding method. Applied Sciences, 2021, 11(24): 11777.

[5] JAIN M, SAIHJPAL V, SINGH N, et al. An overview of variants and advancements of PSO algorithm. Applied Sciences, 2022, 12(17): 8392.

[6] XU K, CHEN Y, ZHANG X, et al. Improved sparrow search algorithm based on multistrategy collaborative optimization performance and path planning applications. Processes, 2024, 12(12): 2775.

[7] XUE J, SHEN B. A novel swarm intelligence optimization approach: sparrow search algorithm. Systems Science & Control Engineering, 2020, 8(1): 22-34.

[8] ZHANG Z, HE R, YANG K. A bioinspired path planning approach for mobile robots based on improved sparrow search algorithm. Advances in Manufacturing, 2022, 10(1): 114-130.

[9] LIU L, LIANG J, GUO K, et al. Dynamic path planning of mobile robot based on improved sparrow search algorithm. Biomimetics, 2023, 8(2): 182.

[10] CHEN Y D, WANG P L, LIN Z C, et al. Global path planning method by fusion of A-star algorithm and sparrow search algorithm//2022 IEEE 11th Data Driven Control and Learning Systems Conference, August 3-5, 2022, Chengdu, China. New York: IEEE, 2022: 205-209.

[11] HE Y, WANG M. An improved chaos sparrow search algorithm for UAV path planning. Scientific Reports, 2024, 14: 366.

[12] LIU G Z, ZHANG S, MA G J, et al. Path planning of unmanned surface vehicle based on improved sparrow search algorithm. Journal of Marine Science and Engineering, 2023, 11(12): 2292.

[13] GAO B, SHEN W, GUAN H, et al. Research on multistrategy improved evolutionary sparrow search algorithm and its application. IEEE Access, 2022, 10: 62520-62534.

[14] ZHU B, BEDEER E, NGUYEN H H, et al. UAV

trajectory planning in wireless sensor networks for energy consumption minimization by deep reinforcement learning. IEEE Transactions on Vehicular Technology, 2021, 70(9): 9540-9554.
[15] HE L, AOUF N, SONG B. Explainable deep reinforcement learning for UAV autonomous path planning. Aerospace Science and Technology, 2021, 118: 107052.
[16] ZHAO Y, MA Y, HU S. USV formation and path-following control via deep reinforcement learning with random braking. IEEE Transactions on Neural Networks and Learning Systems, 2021, 32(12): 5468-5478.
[17] YANG C, YANG H, ZHU D, et al. Chaotic sparrow search algorithm with manta ray spiral foraging for engineering optimization. Systems Science & Control Engineering, 2023, 11(1): 2249021.
[18] GAO Q, ZHENG J, ZHANG W. Research on optimization of manned robot swarm scheduling based on ant-sparrow algorithm. Journal of Physics: Conference Series, 2021, 2078(1): 012002.
[19] QIN H, SHAO S, WANG T, et al. Review of autonomous path planning algorithms for mobile robots. Drones, 2023, 7(3): 211.
[20] ÖZTÜRK Ü, AKDAĞ M, AYABAKAN T. A review of path planning algorithms in maritime autonomous surface ships: navigation safety perspective. Ocean Engineering, 2022, 251: 111010.
[21] KARUR K, SHARMA N, DHARMATTI C, et al. A survey of path planning algorithms for mobile robots. Vehicles, 2021, 3(3): 448-468.
[22] SHI Y, EBERHART R. A modified particle swarm optimizer//IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence, May 4-9, 1998, Anchorage, AK, USA. New York: IEEE, 1998: 69-73.
[23] SHI Y, YANG X, ZHANG Z. Construction of a new time-space two-grid method and its solution for the generalized burgers' equation. Applied Mathematics Letters, 2024, 158: 109244.
[24] LIU Y, HE W, ZHANG H. Grid: guided refinement for detector-free multimodal image matching. IEEE Transactions on Image Processing, 2024, 33: 5892-5906.
[25] HE Y, HU X, ZHANG J, et al. Tower crane path planning based on improved ant colony algorithm. Journal of Measurement Science and Instrumentation, 2024, 15(4): 509-517.
[26] SHI Y, YANG X. A time two-grid difference method for nonlinear generalized viscous burgers' equation. Journal of Mathematical Chemistry, 2024, 62(6): 1323-1356.

改进麻雀搜索算法的无人水面艇路径规划

于 豪, 王 鑫, 彭 皓*

集美大学 轮机工程学院, 福建 厦门 361000

摘 要: 为了实现无人水面艇(USV)的最优导航, 本文提出了一种基于自适应混合策略的麻雀搜索算法(SSA), 以实现高效可靠的路径规划。首先, 增强适应度函数, 以全面考虑关键路径规划指标, 包括路径长度、转弯角度和导航安全性。为了提高搜索多样性并有效避免过早收敛到局部最优值, 在种群初始化阶段采用了混沌映射, 使得算法能够探索更广泛的解空间。其次, 引入了反向惯性权重机制, 以动态平衡不同迭代中的探索和开发。惯性权重的自适应调整进一步提高了收敛效率, 增强了全局优化性能。此外, 还采用了Cauchy-Gaussian混合更新策略, 将随机性和变化注入搜索过程, 这有助于算法摆脱局部最小值并保持高水平的解多样性, 显著提高了优化过程的稳健性和适应性。仿真实验结果显示, 改进后的SSA始终优于原始SSA、PSO和WMR-SSA等基准算法。相比于上述三种算法, 在模拟环境中, 所提出改进算法的路径长度分别减少了21%, 21%和16%, 在实际海洋仿真条件下, 路径长度分别减少了13%, 15%和11%。实验结果突出了改进SSA的有效性和实用性, 为复杂海洋环境中的智能和自主USV导航提供了一种有效的解决方案。

关键词: 适应度函数; 转向角; 混沌映射; 反向惯性权重; 柯西分布; 正弦混沌映射

引用格式: YU Hao, WANG Xin, PENG Hao. Unmanned surface vehicles path planning with improved sparrow search algorithm. Journal of Measurement Science and Instrumentation, 2025, 16(2): 245-257. DOI: 10.62756/jmsi.1674-8042.2025024