Journal of Marine Science and Application (2026) 25: 1446-1455
https://doi.org/10.1007/s11804-025-00713-5

RESEARCH ARTICLE

Chaos-Based Inertia Weight Particle Swarm Optimization for Path Planning of Unmanned Surface Vehicles

Gongxiang Cui[1], Jie Song[1], Ying Han[1], Haifang Wu[1], Fengyu Zhou[2] and Yugang Wang[1]

Received: 20 April 2025 / Accepted: 29 July 2025
© Harbin Engineering University and Springer-Verlag GmbH Germany, part of Springer Nature 2026

Abstract

Unmanned surface vehicles (USVs) play a vital role in marine construction, such as maritime patrol and search–rescue operations. However, USVs face complex challenges when performing tasks in path planning. An adaptive chaotic-enhanced particle swarm optimization (ACA-PSO) algorithm is presented in this paper to increase the efficiency of static path planning for USVs in intricate maritime environments. The proposed ACA-PSO algorithm addresses the dual challenges of path optimality and navigational safety while considering path length, smoothness, safety, and convergence speed. First, a sine chaotic sequence for initialization is introduced, which enhances global search capabilities and effectively escapes local optima. Second, an adaptive inertia weight decay strategy is proposed to adjust the weight dynamically to enhance convergence efficiency and solution quality. Third, sinusoidal acceleration learning factors are considered to alleviate the limitations of static cognitive and social learning factors. Fourth, a historical velocity weighting strategy is developed to improve trajectory smoothness and convergence robustness. Finally, extensive simulation results confirm that the proposed ACA-PSO algorithm significantly outperforms conventional PSO and other optimization algorithms in path planning effectiveness, convergence efficiency, and operational stability. Specifically, in optimal path distance, the proposed ACA-PSO algorithm achieves 2.4% and 4.9% improvements compared with traditional PSO and simulated annealing (SA) algorithms, respectively. The proposed ACA-PSO algorithm also demonstrates a 29.8% better grayscale value in path smoothness compared with the SA algorithm. Moreover, with a much lower standard deviation of 0.201×10^{-2}, the ACA-PSO algorithm is superior in stability.

Keywords Unmanned surface vehicles (USVs); Particle swarm optimization (PSO); Path planning; Chaos sequences; Adaptive inertia weights

1 Introduction

Chen et al. (2025) and Luo and Su (2025) demonstrated that unmanned surface vehicles (USVs) are extensively applied in military and civilian domains due to their advantages of robust environmental adaptability and superior operational efficiency. However, the efficiency and safety of USVs' path planning are critical in performing tasks, which has attracted growing attention in recent research.

Path planning, a central aspect of USVs, is defined as identifying an optimal navigation route from a starting point to a target destination. Shen et al. (2021) and Li et al. (2024) showed that state-of-the-art research on related algorithms has developed substantially over the past decades. Cui et al. (2025) established the QuatAPF framework, combining multiple potential fields to ensure collision-free, energy-efficient operations for deep-sea mining vehicles. Khattab et al. (2023) developed an RRT-based approach that effectively handles path planning in complex high-dimensional spaces. Huang and Guo (2022) enhanced the A* algorithm to improve computational efficiency in large-scale static environments. Aashamri et al. (2022) optimized Dijkstra's algorithm for real-time pathfinding in structured grids. Similarly, Liao et al. (2025) demonstrated the robustness of potential field methods for obstacle avoidance in stationary terrains. Although these algorithms are effective in structured or static scenarios, their adaptability

Article Highlights

- A sine chaotic sequence is introduced to enhance population diversity and to prevent the presented algorithm from becoming trapped in local optima.
- An adaptive inertia weight decay strategy is implemented to strengthen the balance between global exploration and local exploitation and to improve convergence performance.
- A historical velocity weighting mechanism is introduced to incorporate past velocities and to improve trajectory smoothness and algorithm stability.

✉ Yugang Wang
sdu201413070@gmail.com

[1] College of Naval Architecture and Marine Engineering, Shandong Jiaotong University, Weihai 264200, China

[2] School of Control Science and Engineering, Shandong University, Jinan 250061, China

is limited when applied to complex marine environments with irregular obstacles and nonuniform terrain conditions.

Lin et al. (2025), Gronauer and Diepold (2022), and Zhou et al. (2025) revealed that reinforcement learning (RL) and deep reinforcement learning (DRL) have emerged as powerful optimization methods for USV path planning and enable learning optimal navigation policies through interaction with dynamic marine environments. For cooperative path planning of USVs, Wen et al. (2023) proposed a two-stage DRL solution that complies with COLREGs and effectively manages collision avoidance within the USV fleet and between USVs and target ships. To traverse USVs in environments with partial observations, Xiao et al. (2023) implemented a DRL approach that combines deep Q-networks with long short-term memory and improves path planning efficiency and adaptation to unknown information. For multiUSV task planning, Zhang et al. (2024) established an enhanced DRL approach that divides the task planning problem into autonomous collision avoidance and job allocation subproblems. However, Wang et al. (2023a) presented that RL generally necessitates substantial interaction samples for effective strategy learning. Moreover, Zhao et al. (2021) and Yan et al. (2024) showed that RL algorithms commonly converge to local optima and often fail to identify globally optimal navigation paths.

Lei et al. (2023) and Gul et al. (2022) demonstrated the superior performance of metaheuristic algorithms in handling complex combinatorial optimization problems compared with traditional methods, which makes them principally valuable for USV path planning research. Cleghorn and Stapelberg (2022) and Zhang and Lin (2022) showed that particle swarm optimization (PSO) has emerged as a promising solution due to its simple structure, rapid convergence properties, and inherent parallelism that facilitates distributed computing implementations, as proven by Campos et al. (2019). The state of the art focused on strengthening PSO performance through several strategies. In terms of initialization mechanisms, to address the slow convergence caused by insufficient population distribution in evolutionary algorithms, Chu et al. (2022) proposed a chaotic initialization method based on Logistic maps and effectively accelerated optimization progress while maintaining solution diversity. Similarly, to overcome the performance degradation of traditional PSO in high-dimensional search spaces, Yang et al. (2022) proposed a differential elite learning PSO algorithm for global numerical optimization and enhanced optimization performance in complex scenarios. Moreover, Zare et al. (2023) proposed a modified PSO algorithm considering hummingbird flight patterns to improve search quality and population and solution quality in engineering applications. The algorithm's dynamic adjustment capabilities have been remarkably improved through nonlinear inertia weight adjustment for better exploration-exploitation balance (Zhang and Li, 2024) and time-varying acceleration coefficients for adaptive parameter control (Ratnaweera et al., 2004). Despite these advances, critical challenges remain unresolved. Westheider et al. (2023) addressed multiagent path planning, but their approach did not incorporate comprehensive parameter adaptation. Li et al. (2020) improved velocity-driven strategies, but their method displayed constraints in maintaining search diversity. Mozzen et al. (2023) improved memory mechanisms, but their framework struggled to achieve robust convergence in complex environments. Qiu et al. (2023), Sun (2024), and Nguyen et al. (2024) showed that despite the substantial progress in specific aspects of USV path planning, a critical gap remains: the absence of a unified framework that can address all key challenges in this domain.

To overcome these limitations, an enhanced adaptive chaotic accelerated PSO (ACA-PSO) algorithm is proposed in this study. The foremost contributions include implementation of sine chaotic sequence initialization for improved global search capability, development of adaptive inertia weight strategy for dynamic balance, introduction of sinusoidal acceleration learning factors for optimized cognitive-social interaction, and integration of historical velocity weighting mechanism for enhanced convergence stability. The experimental results present that ACA-PSO achieves superior performance in path planning effectiveness, convergence efficiency, and operational stability compared with existing approaches.

The remaining parts of this paper are organized as follows. Section 2 develops the mathematical model for USV path planning. Section 3 discusses the design methodology and implementation of the enhanced ACA-PSO algorithm. Section 4 confirms the algorithm's efficacy through comparative simulations. Section 5 presents the conclusions.

2 Problem description

Surface motion path planning for USVs is investigated in this paper, as exhibited in Figure 1. The main objective is to determine the optimal path from the initial position $S = (x_0, y_0)$ to the target position $S_n = (x_n, y_n)$ while optimizing path smoothness at turning points and guaranteeing safety performance. Path quality is assessed through a comprehensive fitness function that integrates numerous objectives. The USV's motion characteristics are defined by its velocity v_i and turning angle φ, with corresponding velocity components v_x and v_y along the x and y axes, respectively.

2.1 Path length calculation

Obtaining a better path length enables reducing the overall computing overhead and running time of USV during execution. Total path length f_{length} is a primary optimization

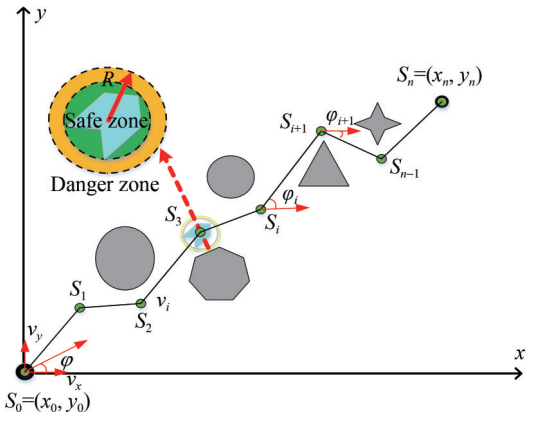


Figure 1 Schematic diagram of USV path planning on the sea surface

objective in USV navigation. Based on the Euclidean distance metric, the cumulative length of a path composed of n way points is computed as follows.

$$f_{\text{length}} = \sum_{i=0}^{n-1} \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2} \tag{1}$$

where (x_i, y_i) and (x_{i+1}, y_{i+1}) are the Cartesian coordinates of the ith and $(i+1)$th way points, respectively, and n is the total number of path segments. This formula guarantees a geometrically precise representation of the USV's travel distance. In the path planning of unmanned ships, the optimization of the path length not only signifies a reduction in the navigation distance but is also more directly related to the overall calculation and operation costs.

2.2 Path smoothness complexity evaluation

During navigation, USVs often encounter sharp directional changes. To improve path smoothness, a quantitative smoothness evaluation function f_{smooth} is proposed. This function evaluates path smoothness by calculating the turning angles between consecutive path segments.

Let the directional vectors of two consecutive path segments, denoted as the ith and $(i+1)$th segments, be represented by $(\mathrm{d}x_i, \mathrm{d}y_i)$ and $(\mathrm{d}x_{i+1}, \mathrm{d}y_{i+1})$, respectively.

$$\theta_i = \left| \arctan\left(\mathrm{d}y_{i+1}, \mathrm{d}x_{i+1}\right) - \arctan\left(\mathrm{d}y_i, \mathrm{d}x_i\right) \right| \tag{2}$$

where $\varphi = \arctan\left(\mathrm{d}y, \mathrm{d}x\right)$ is the angle between the vector $\left(\mathrm{d}x, \mathrm{d}y\right)$ and the positive x axis, measured in radians. Smoothness complexity evaluation function f_{smooth} is expressed as follows.

$$f_{\text{smooth}} = \frac{1}{\pi} \sum_{i=2}^{N-1} \left(\theta_i \times \frac{1}{v_i} \right) \times 100 \tag{3}$$

where the path comprises N segments (yielding $N-1$ turning angles), $\sum_{i=2}^{N-1} \theta_i$ is the summation of all turning angles along the path, v_i is the velocity of the ith path segment, and $1/\pi$ is a normalization coefficient that scales the result to the range [0, 100], quantifying the path complexity.

2.3 Safety value evaluation metric

To safeguard collision-free navigation, a safety zone with radius R is established around the USV path centerline, as presented in Figure 1. This zone functions as a protective buffer against obstacles. Safety evaluation function f_{safe} quantifies path feasibility through the following formula.

$$f_{\text{safe}} = \begin{cases} 1, & d_m \geqslant R \\ 0, & d_m < R \end{cases} \tag{4}$$

where d_m is the minimum distance between the USV's path centerline and the mth obstacle. When $d_m < R$, an intersection exists between the safety zone and obstacles, and the evaluation function returns zero, rendering the planned path infeasible. Conversely, when $d_m \geqslant R$, the path satisfies the safety constraints and is deemed feasible.

2.4 Fitness function for path planning

By integrating all the said factors and their corresponding evaluation metrics, the comprehensive fitness function f for path planning is expressed through the synthesis of Equations (1) to (4), as follows.

$$f = \frac{1}{f_{\text{length}}} \times \frac{1}{f_{\text{smooth}}} \times f_{\text{safe}} \tag{5}$$

Equation (5) presents that path length, smooth, safety, and convergence speed are involved in the path planning of USV, which can reduce computational complexity. A higher fitness value indicates superior path planning performance, characterized by optimal path length, enhanced smoothness, and improved safety margins. Conversely, a lower fitness value reflects suboptimal performance, manifested in excessive path length, increased complexity, and compromised safety parameters.

3 Improvement of the PSO algorithm

The ACA-PSO algorithm has been developed based on the traditional PSO algorithm and incorporates a sine chaotic sequence for initialization, an adaptive inertia weight decay strategy, sinusoidal acceleration learning factors, and a historical velocity weighting mechanism to enhance path planning performance. The improvement strategies and algorithmic implementation are detailed as follows.

3.1 PSO algorithm

The PSO algorithm employs iterative equations to update particle velocities and positions and direct the swarm toward optimal solutions through the exploitation of personal best positions (P^{best}) and the global best position (G^{best}). Canonical velocity and position update equations are expressed as follows (Kennedy and Eberhart, 1995).

$$v_i(t+1) = wv_i(t) + c_1 r_1 (P^{\text{best}_i} - x_i(t)) + c_2 r_2 (G^{\text{best}} - x_i(t)) \tag{6}$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \tag{7}$$

where $v_i(t+1)$ is the velocity vector of the ith particle at iteration $t+1$; $x_i(t+1)$ is the position vector of the ith at iteration $t+1$; c_1 and c_2 are the cognitive and social acceleration coefficients, respectively, which regulate the particle's velocity adjustment based on personal and global experience; r_1 and r_2 are random variables uniformly distributed in the interval $[0, 1]$; and w is the inertia weight coefficient that modulates the effect of the current velocity on the subsequent position update.

3.2 Improved ACA-PSO algorithm

3.2.1 Sine chaotic sequence

To improve population diversity, a sine chaotic mapping replaces the standard random initialization and generates pseudorandom sequences with ergodicity and nonrepetition. Chaotic weight w is defined as follows (Huang et al., 2025).

$$w = 0.5 \times \left(1 + \frac{k}{5} \times \sin(\pi y_i)\right) \tag{8}$$

where t is the iteration index, y_i is a stochastic variable uniformly distributed in the interval (0, 1.5), and k is the control coefficient. Figure 2 illustrates the temporal evolution of w over 60 iterations. For the specific case where $y_i = 0.15$ and $k = 5$, w exhibits a nonlinear dynamic behavior bounded within [0, 1]. This proposed sine-based dynamic adaptation mechanism not only provides effective modulation of the inertia weight but also improves the stochastic nature and population diversity of the search through the incorporation of the stochastic parameter.

3.2.2 Adaptive inertia weight decay strategy

To balance global exploration and local exploitation dynamically, an adaptive inertia weight w, which decays nonlinearly with iterations, is proposed (Wang et al., 2023b). The weight update mechanism is defined as follows.

$$w = w_{\max} + (w_{\max} - w_{\min}) \times \Phi \tag{9}$$

where Φ is a monotonically decreasing function of the current iteration index k and the maximum number of iterations T, expressed as follows (Zhou et al., 2022).

$$\Phi = \cos\left(\left(\frac{k}{T}\right)^3 \times \frac{\pi}{2}\right) \tag{10}$$

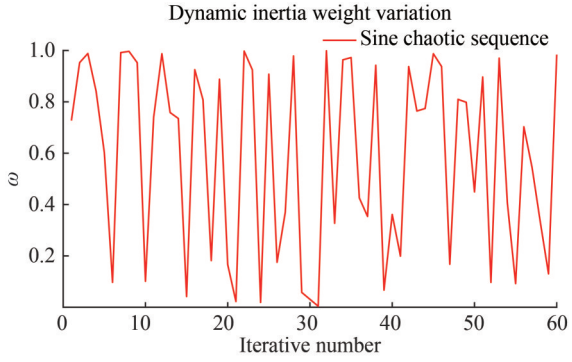


Figure 2 Dynamic variation of the inertia weight w using the sine chaotic sequence over 60 iterations

The proposed adaptive strategy implements a larger inertia weight during early iterations to facilitate extensive exploration of the search space while progressively decreasing the weight in later iterations to intensify local exploitation. Weight dynamic adjustment mechanism effectively improves the algorithm's convergence characteristics and optimization performance.

3.2.3 Sine acceleration learning factor

To balance exploration and exploitation adaptively, the cognitive c_1 and c_2 coefficients are dynamically modulated via sinusoidal functions over iterations (Zhan et al., 2009).

$$c_1 = A \times \cos\left(\left(1 - \frac{k}{T}\right) \times \frac{\pi}{3}\right) + B \tag{11}$$

$$c_2 = A \times \sin\left(\left(1 - \frac{k}{T}\right) \times \frac{\pi}{3}\right) + B \tag{12}$$

where A and B are the control parameters, k is the current iteration index, and T is the maximum number of iterations. During early iterations, larger values of c_1 and c_2 enhance the particles' global exploration capabilities and facilitate comprehensive coverage of the solution space. As iterations progress, the decremental values of c_1 and c_2 intensify local exploitation and enable particles to refine potential solutions, thus optimizing convergence rate and solution precision.

3.2.4 Historical velocity weighting strategy

To improve the algorithm's stability and to accelerate convergence, this study introduces an innovative optimization strategy incorporating historical velocity weighting. The proposed approach integrates a weighted average of historical velocities into the velocity update equation and

modifies the conventional update formula.

$$v_i(t+1) = wv_i(t) + c_1 r_1\left(P^{\text{best}_i} - x_i(t)\right) + c_2 r_2\left(G^{\text{best}} - x_i(t)\right) + \alpha \cdot \text{Avg}\left[\sum_{k=1}^{5} v_i(t-k)\right] \tag{13}$$

where $\alpha = 0.1$ is the historical velocity weighting coefficient, $\sum_{k=1}^{5} v_i(t-k)$ is the aggregation of individual particle velocities over five consecutive time steps (from $t-1$ to $t-5$), and Avg is the weighted average velocity calculated from the five earlier iterations. The velocity weighting enhancement remarkably strengthens the algorithm's temporal memory capacity, alleviates oscillatory behavior during optimization, and simultaneously improves convergence stability and solution precision.

3.3 Specific steps of the ACA-PSO algorithm

The proposed ACA-PSO algorithm integrates four key enhancement strategies: 1) sine chaotic initialization, 2) adaptive inertia weight decay, 3) sinusoidal acceleration learning factors, and 4) historical velocity weighting. The complete algorithmic procedure is expressed in Table 1, and the implementation details are as follows.

Table 1 ACA-PSO algorithm

Require: c_1, c_2, acceleration coefficients; N, swarm size; D, swarm dimension; k, control factor for Sine chaos; α, historical velocity weighting coefficient; Max_iter, maximum number of iterations.

Process

Initialize the particles of the swarm;
  Initialize the path planning environment and constraints;
  Set parameters, N, Max_iter c_1, c_2;
  for each particle i in swarm **do**;
    Initialize position x_i at starting point (0, 0);
    Initialize velocity v_i using Sine chaotic sequence /* Equation (8) */
    Set global_best G = argmin $\{f(P_i)\}$;
  end for
  while iter < Max_iter **do**
    Calculate adaptive inertia weight /* Equation (9) */
    for each particle i in swarm **do**
      if particle is trapped in local optima **then**
      Apply chaos perturbation to escape local optima;
      Else
    Update particle velocity v_i using/* Equation (6) */with historical velocity weighting /* Equation (13) */
      Update particle position x_i using /* Equation (7) */
      Calculate new fitness $f(x_i)$;
      if $f(x_i) < f(p_i)$ **then**
        $p_i = x_i$;
      end if
      if $f(p_i) < f(G)$ **then**
        $G = P_i$
      end if
    end if
  end for
    Update historical velocity weights /* Equation (13) */
    if convergence criterion is met **then**
      break;
    end if
  end while

Return Optimal path G and fitness $f(G)$ as the final optimal solution.

4 Simulation experiments and analysis

Comprehensive experimental validation is discussed in this section to assess the performance of the ACA-PSO algorithm through extensive numerical simulations and comparative analyses.

4.1 Parameter configuration

The experimental evaluation was performed in the MATLAB R2024a environment on a computing platform equipped with an Intel Core i7 processor and 16 GB of RAM. Simulation experiments were performed to assess the performance of the USV path planning algorithm. Comparative analyses were completed against traditional algorithms, including PSO, genetic algorithm (GA), simulated annealing (SA), and ant colony optimization (ACO). The key parameters were configured as follows: The population size for all five algorithms was uniformly set to 50, while the maximum number of iterations was fixed at 1 000. Each algorithm was independently executed 30 times on the same test function, and the optimal value (Best), mean value (Mean), and standard deviation (Std) of the results from these 30 runs were calculated. For the proposed ACA-PSO algorithm, the initial inertia weight ($w_{\max}$) was set to 0.9, the minimum inertia weight ($w_{\min}$) was set to 0.4, the acceleration factors (A and B) were set to 2.0 and 1.5, respectively, and the historical velocity weighting coefficient α was set to 0.1.

4.2 Path planning for USVs with regular obstacles

To validate the performance of the proposed ACA-PSO algorithm in a basic scenario, initial experiments were com-

pleted in an environment featuring multiple circular obstacles. The experimental setup included a starting point $S = (0, 0)$, a target point $T = (7, 9.1)$, and six circular obstacles of varying sizes.

Figure 3 illustrates the path planning processes and results of the five algorithms (PSO, SA, GA, ACO, and ACA-PSO). Figure 3 reveals that the proposed ACA-PSO algorithm outperforms the other algorithms, not only achieving effective path planning but also avoiding all obstacles successfully. Additionally, the proposed ACA-PSO algorithm demonstrates superior path smoothness and a shorter path length compared with the alternatives.

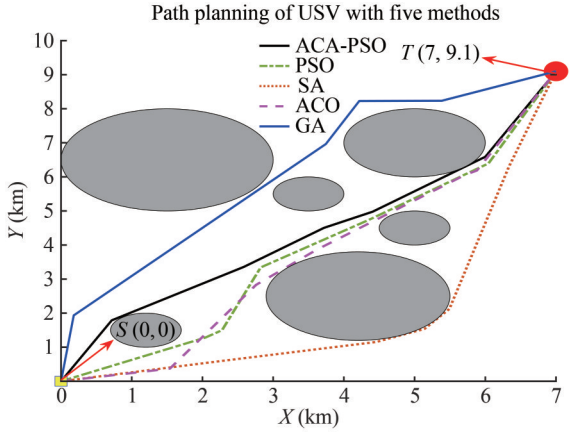


Figure 3 Path planning process of five algorithms under regular path conditions

To demonstrate better the convergence speed and efficiency of the proposed strategy, Figure 4 presents the fitness convergence curves of traditional PSO and improved ACA-PSO algorithms during iteration. The ACA-PSO algorithm achieves stability after approximately 200 iterations, whereas the traditional PSO algorithm requires up to 800 iterations to converge. Moreover, the final fitness value of ACA-PSO is substantially higher than that of traditional PSO. The experimental result indicates that the proposed improvement strategy considerably enhances the convergence speed and optimization capability of the algorithm.

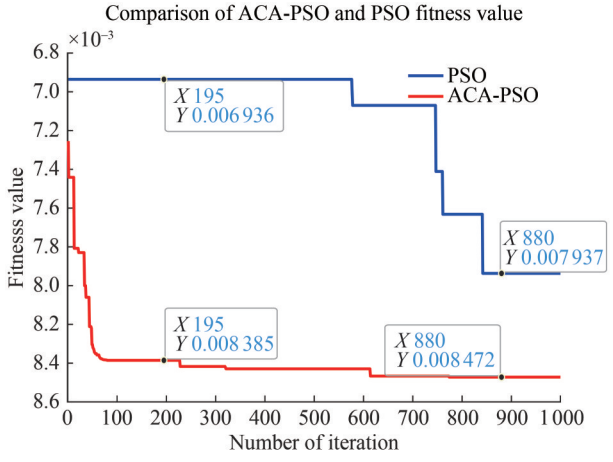


Figure 4 Evolution curve of the optimal individual fitness

To evaluate the performance of the ACA-PSO algorithm quantitatively compared with other algorithms, this paper conducted a comprehensive analysis focusing on four key metrics: optimal path distance, average convergence speed, fitness value/100, and execution time. Table 2 details the results from 30 independent runs of the five algorithms.

The experimental data analysis in Table 2 shows that the proposed ACA-PSO algorithm demonstrates remarkable advantages across multiple evaluation metrics. In terms of the core metric of path planning, that is, optimal distance, the ACA-PSO algorithm achieved a best value of 11.666 89, representing improvements of 2.4% and 4.9% over the traditional PSO and SA algorithms, respectively. Furthermore, the ACA-PSO algorithm's Std of 0.201×10^{-2} was substantially lower than those of the other comparative algorithms, indicating superior stability. Regarding path smoothness, the ACA-PSO algorithm achieved a grayscale value of 0.248 854, outperforming the SA algorithm's value of 0.354 53 by 29.8%, which demonstrates that the paths generated by ACA-PSO exhibit enhanced continuity and executability. Although the ACO algorithm achieved the best

Table 2 Performance comparison of five algorithms in an environment with circular obstacles

Metrics	Index	PSO	SA	GA	ACO	ACA-PSO
Optimal path distance	Std	0.316 12	0.164 05	0.168 22	0.013 73	$\mathbf{0.201 \times 10^{-2}}$
	Mean	12.550 2	13.083 2	12.942 4	12.252 8	**11.714 79**
	Best	11.954 2	12.265 3	12.453 2	12.027 2	**11.666 89**
Average convergence speed	Std	$\mathbf{6.727 \times 10^{-14}}$	6.564×10^{-6}	$1.762\,7 \times 10^{-6}$	0.242×10^{-5}	$8.365\,1 \times 10^{-14}$
	Mean	7.341×10^{-7}	0.005 45	$1.762\,7 \times 10^{-6}$	**0.013 14**	$1.147\,6 \times 10^{-6}$
	Best	1.186×10^{-6}	0.009 14	$1.762\,7 \times 10^{-6}$	**0.014 98**	$1.581\,7 \times 10^{-6}$
Fitness value/100	Std	0.186 42	0.002 79	0.040 63	0.161 76	$\mathbf{0.509 \times 10^{-2}}$
	Mean	0.818 91	0.445 41	0.682 06	0.745 98	**0.333 725**
	Best	0.370 53	0.354 53	0.483 29	0.673 87	**0.248 854**
Execution time	Std	0.126 4	**0.064 3**	0.912 3	1.095 9	0.769×10^{-1}
	Mean	7.036 4	**1.286 6**	2.966 5	60.753 2	4.699 52
	Best	6.605 3	**1.099 8**	1.779 2	59.169 4	3.589 95

Note: Bold values indicate the best performance for each metric.

average convergence speed of 0.013 14 and the SA algorithm recorded the fastest runtime of 1.099 8 s, the ACA-PSO algorithm maintained competitive performance in these secondary metrics. Notably, the ACA-PSO algorithm exhibited smaller fluctuations in Stds across all metrics, highlighting its stable, reliable performance under varying operating conditions. These experimental results validate the effectiveness and optimization performance of the proposed algorithm in solving path planning problems.

4.3 Path planning for USVs with irregular obstacles

To further validate the effectiveness and advancement of the ACA-PSO algorithm in complex environments, this experiment replaced circular obstacles with irregular polygons to approximate real-world conditions more closely. Figure 5 illustrates the path planning processes and results of the five algorithms in an environment featuring irregular obstacles.

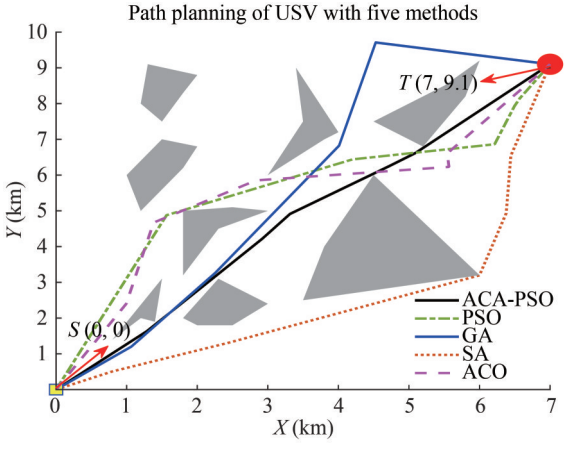


Figure 5 Path planning process of five algorithms under irregular path conditions

Figure 5 shows that the trajectory generated by the ACA-PSO algorithm exhibits exceptional environmental adaptability. The algorithm demonstrates superior performance in two critical aspects: robust obstacle avoidance capability for irregular-shaped obstacles and optimal trajectory smoothness characteristics. The generated path effectively navigates through complex environmental constraints while maintaining continuous, smooth motion characteristics.

Figure 6 compares the convergence processes between the ACA-PSO and traditional PSO algorithms. Even in a more complex environment, the ACA-PSO algorithm retains its rapid, stable convergence characteristics and achieves stability after approximately 200 iterations.

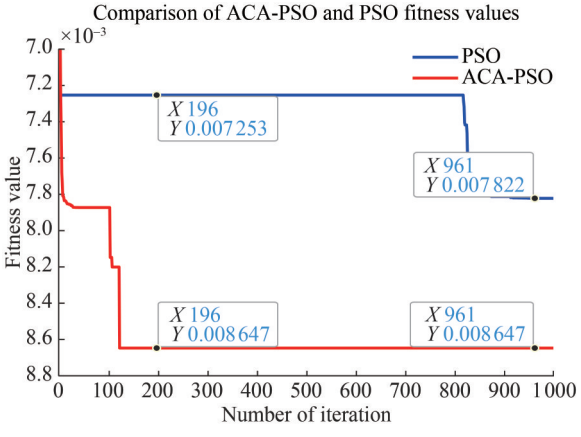


Figure 6 Evolution curve of the optimal individual fitness

To further demonstrate the superiority of the proposed algorithm, Table 3 details quantitative performance metrics, including optimal path distance, average convergence speed, fitness value/100, and execution time.

Experimental data demonstrate that the proposed ACA-PSO algorithm maintains substantial performance advantages even in complex environments. In terms of the key metric of path planning, namely, optimal distance, the ACA-PSO algorithm achieved a best value of 11.487 33, representing a 2.2% improvement over the 11.739 achieved by the PSO algorithm and considerably outperforming the other comparative algorithms. Notably, the ACA-PSO algo-

Table 3 Performance comparison of five algorithms in an environment with irregular obstacles

Metrics	Index	PSO	SA	GA	ACO	ACA-PSO
Optimal path distance	Std	0.177 1	**0.783×10^{-4}**	0.446 1	0.307×10^{-3}	0.828×10^{-2}
	Mean	12.581	12.841	13.328	12.642 2	**11.580 04**
	Best	11.739	12.779	12.456	12.600 4	**11.487 33**
Average convergence speed	Std	7.049×10^{-14}	1.01×10^{-5}	8.362×10^{-10}	8.473×10^{-6}	**$5.090\,9\times10^{-14}$**
	Mean	9.137×10^{-7}	0.006 7	4.766×10^{-5}	**0.008 5**	$1.801\,9\times10^{-6}$
	Best	1.375×10^{-6}	0.012 4	8.561×10^{-5}	**0.013 7**	$2.093\,4\times10^{-6}$
Fitness value/100	Std	0.103 88	**0.028×10^{-1}**	0.117 2	0.043 5	0.967×10^{-1}
	Mean	0.498 3	**0.401 3**	0.958 9	0.871 2	0.660 74
	Best	0.319 3	0.331 3	0.616 1	0.607 9	**0.117 69**
Execution time	Std	0.209×10^{-1}	0.027 1	0.508 8	0.843 7	**0.207×10^{-1}**
	Mean	2.974 7	2.416 6	3.328 2	125.61	**1.624 8**
	Best	2.875 8	2.187 1	2.174 6	124.19	**1.312 4**

Note: Bold values indicate the best performance for each metric.

rithm exhibited exceptional computational efficiency, with an optimal runtime of only 1.312 4 s, a remarkable 98.8% reduction compared with the 124.19 s required by the ACO algorithm, underscoring its high efficiency in practical applications. Regarding algorithm stability, the ACA-PSO algorithm achieved Stds of (0.028×10^{-1}) for optimal distance and (0.207×10^{-1}) for runtime, exhibiting excellent stability in the two performance critical metrics. Although the SA algorithm achieved a slightly better Std in path smoothness (0.028×10^{-1}) compared with the ACA-PSO algorithm (0.967×10^{-1}), the ACA-PSO algorithm's optimal grayscale value of 0.117 69 was significantly superior to those of the PSO algorithm (0.319 3) and the SA algorithm (0.331 3), underscoring its advantage in path smoothness. These experimental results validate that the ACA-PSO algorithm achieves balanced performance across multiple dimensions, including path optimization quality, computational efficiency, and algorithm stability, when addressing path planning problems in complex environments.

4.4 Path planning for USVs with simulation virtual environment

To further validate the performance of ACA-PSO, the virtual environment, which contains multiple irregular static obstacles, was constructed. The experimental setup included six fixed obstacle zones with varying shapes and sizes, which were designed to simulate diverse topographic barriers that may be encountered in real-world aquatic environments.

The results in Figure 7 compare the path planning performance of the five algorithms (ACA-PSO, ACO, PSO, GA, and SA) in a complex irregular environment. The ACA-PSO algorithm (solid black line) demonstrates strong performance, generating a smooth, obstacle-free trajectory with minimal turns and high efficiency. The performance result highlights ACA-PSO's ability to balance precision, smoothness, and efficiency effectively in challenging scenarios. By contrast, the ACO algorithm (purple dash-dotted line) achieves basic obstacle avoidance but exhibits oscillations in narrow regions, which influences path stability. The PSO (dashed green line) and GA (solid blue line) algorithms face challenges with excessive directional adjustments, resulting in convoluted paths that lack practical efficiency. The SA algorithm (red dotted line) struggles the most, often becoming trapped in local optima and producing redundant detours with sharp turns. Overall, ACA-PSO stands out as a robust, efficient method in this complex irregular environment and delivers a superior balance of performance metrics.

Figure 8 presents a comparative analysis of fitness values between the ACA-PSO algorithm and the conventional PSO algorithm during iteration. The convergence curves demonstrate that the ACA-PSO algorithm achieves a favorable convergence state (fitness value ≈ 0.008 315 67) after approximately 200 iterations, whereas the conventional

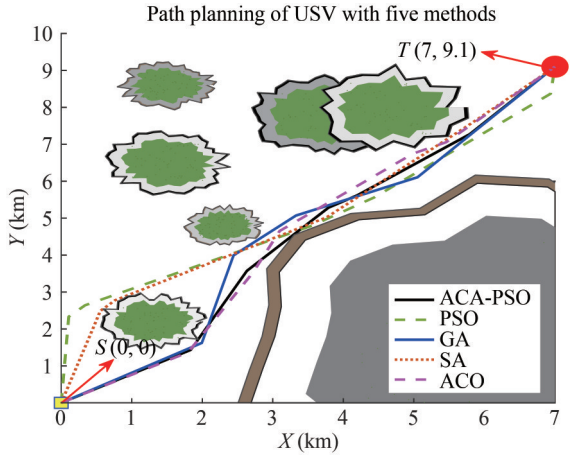


Figure 7 Path planning of five algorithms under intricacy path conditions

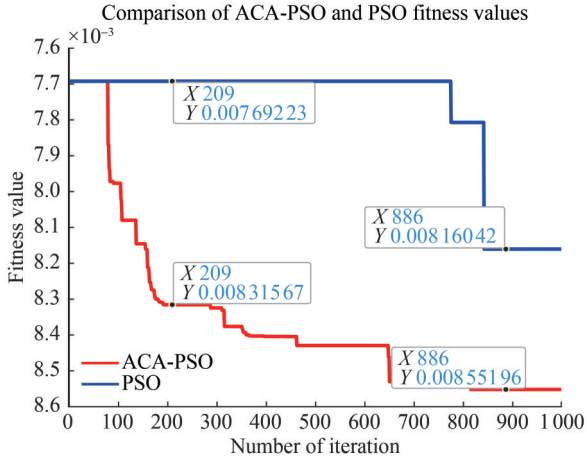


Figure 8 Evolution curve of the optimal individual fitness

PSO algorithm needs almost 900 iterations to attain a comparable convergence level (fitness value = 0.008 160 42). These results indicate that the ACA-PSO algorithm exhibits superior convergence speed and enhanced optimization capability.

The experimental results in Table 4 demonstrate that the ACA-PSO algorithm achieves optimal performance in path distance (optimal path distance = 11.693 25) and comprehensive fitness (fitness value = 0.721 99), outperforming conventional PSO by a 2.45% reduction in path length while exhibiting enhanced obstacle avoidance capability in dense obstacle scenarios. Although the SA algorithm demonstrates substantially faster average convergence speed (0.007 2) compared with ACA-PSO ($8.163\,5 \times 10^{-7}$), indicating superior search efficiency, ACA-PSO exhibits markedly more stable convergence characteristics, as indicated by its ultralow Std of $1.416\,8 \times 10^{-15}$. Furthermore, ACA-PSO achieves the shortest execution time (6.253 9 s) among comparable algorithms (e.g., GA's 23.256 s), demonstrating substantial improvements in computational efficiency while meeting real-time requirements. The ACO algorithm

Table 4 Performance comparison of four algorithms in a simulation virtual environment

Metrics	Index	PSO	SA	GA	ACA-PSO
Optimal path distance	Std	0.10822	0.07277	0.07131	$\mathbf{0.243\times10^{-3}}$
	Mean	12.4602	12.1398	12.1409	**11.70816**
	Best	11.9865	11.8594	11.7936	**11.69325**
Average convergence speed	Std	$\mathbf{2.032\times10^{-16}}$	2.57×10^{-6}	5.804×10^{-13}	1.4168×10^{-15}
	Mean	4.309×10^{-7}	**0.0072**	7.625×10^{-6}	8.1635×10^{-7}
	Best	4.686×10^{-7}	**0.0098**	7.936×10^{-6}	8.6854×10^{-7}
Fitness value/100	Std	0.769×10^{-1}	0.529×10^{-2}	0.234×10^{-2}	$\mathbf{0.458\times10^{-3}}$
	Mean	2.7562	**0.3192**	0.5462	0.74414
	Best	2.5425	**0.2365**	0.4936	0.72199
Execution time	Std	0.15484	0.2298	1.2694	$\mathbf{0.322\times10^{-1}}$
	Mean	12.2086	6.8321	25.052	**6.4881**
	Best	11.7317	6.1771	23.256	**6.2539**

Note: Bold values indicate the best performance for each metric.

was excluded from the performance comparison due to its impractical execution time (exceeding 3 h) in such complex environments. These algorithmic performance differences stem from distinct algorithmic design emphases: SA prioritizes rapid search capabilities, whereas ACA-PSO excels in path optimization precision and stability.

Above all, the experimental results conclusively validate the effectiveness of the proposed improvement strategies. By incorporating the sine chaotic sequence, adaptive inertia weight adjustment mechanism, and historical velocity weighting strategy, the ACA-PSO algorithm demonstrates remarkable advantages across three distinct environmental complexities.

5 Conclusions

This paper presents an enhanced ACA-PSO algorithm to address key challenges in path planning for USVs. The proposed ACA-PSO algorithm incorporates several improvements: initializing the population with a sine chaotic sequence to enhance global search capability, adopting an adaptive inertia weight decay strategy to balance exploration and exploitation, utilizing sinusoidal acceleration learning factors to optimize cognitive and social interactions, and applying a historical velocity weighting mechanism to ensure smoother trajectories during optimization.

The simulation results demonstrate that the ACA-PSO algorithm achieves remarkable enhancements in path length, convergence speed, computational efficiency, and path smoothness, exhibiting robust environmental adaptability and path planning capability. This study provides reliable technical support for the practical implementation of USV autonomous navigation systems.

Competing interests The authors have no competing interests to declare that are relevant to the content of this article.

References

Aashamri A, Boubacker S, Kolsi L (2022) Improved Dijkstra algorithm for mobile robot path planning and obstacle avoidance. Computers, Materials & Continua 72: 5931-5946. https://doi.org/10.32604/cmc.2022.028165

Campos A de, Pozo A TR, Duarte EP (2019) Parallel multi-swarm PSO strategies for solving many objective optimization problems. Journal of Parallel and Distributed Computing 126: 13-33. https://doi.org/10.1016/j.jpdc.2018.11.008

Chen X, Yin S, Li Y, Xiang Z (2025) Dynamic path planning for multi-USV in complex ocean environments with limited perception via proximal policy optimization. Ocean Engineering 326: 120907. https://doi.org/10.1016/j.oceaneng.2025.120907

Chu H, Yi J, Yang F (2022) Chaos particle swarm optimization enhancement algorithm for UAV safe path planning. Applied Sciences 12: 8977. https://doi.org/10.3390/app12188977

Cleghorn CW, Stapelberg B (2022) Particle swarm optimization: stability analysis using N-informers under arbitrary coefficient distributions. Swarm and Evolutionary Computation 71: 100060. https://doi.org/10.1016/j.swevo.2022.100060

Cui C, Wu S, Chen X (2025) QuatAPF: Safety-aware and energy-efficient dynamic path planning and tracking control for deep-sea mining vehicles via a quatre-artificial potential field method. Ocean Engineering 321: 120236. https://doi.org/10.1016/j.oceaneng.2024.120236

Gronauer S, Diepold K (2022) Multi-agent deep reinforcement learning: A survey. Artificial Intelligence Review 55: 895-943. https://doi.org/10.1007/s10462-021-09996-w

Gul F, Mir I, Aliabadi L, Mir S (2022) Implementation of bio-inspired hybrid algorithm with mutation operator for robotic path planning. Journal of Parallel and Distributed Computing 169: 171-184. https://doi.org/10.1016/j.jpdc.2022.06.014

Huang W, Li W, Pan X, Liu Q, Yang J (2025) Enhanced particle swarm optimization with chaotic search for offshore micro-energy systems. Scientific Reports 15: 1191. https://doi.org/10.1038/s41598-025-85557-3

Huang Y, Guo S (2022) Path planning of mobile robots based on improved A* algorithm. 2022 Asia Conference on Advanced Robotics, Automation, and Control Engineering (ARACE) IEEE, Qingdao, China, 1-5. https://doi.org/10.1109/ARACE56528.2022.00031

Kennedy J, Eberhart R (1995) Particle swarm optimization. Proceedings of ICNN'95-International Conference on Neural

Networks, IEEE, Perth, Australia, 1942-1948. https://doi.org/10.1109/ICNN.1995.488968

Khattab O, Yasser A, Jaradat MA, Romdhane L (2023) Intelligent adaptive RRT path planning algorithm for mobile robots. 2023 Advances in Science and Engineering Technology International Conferences (ASET), IEEE, Dubai, United Arab Emirates, 1-5. https://doi.org/10.1109/ASET5582.2023.1018074

Lei T, Sellers T, Luo C, Carruth DW, Bi Z (2023) Graph-based robot optimal path planning with bio-inspired algorithms. Biomimetic Intelligence and Robotics 3: 10019. https://doi.org/10.1016/j.birob.2023.10019

Li C, Yao L, Mi C (2024) Fusion algorithm based on improved A and DWA for unmanned surface vehicle path planning. Journal of Marine Science and Application 24: 224-237. https://doi.org/10.1007/s42452-024-02237-8

Li W, Fan Y, Jiang Q, Xu Q (2020) Velocity-driven particle swarm optimization. Proceedings of the 2019 8th International Conference on Computing and Pattern Recognition (ICCPR'19), New York, United States, 1-5. https://doi.org/10.1145/3373509.3373517

Liao Y, Tang X, Chen C, Ren Z, Pang S, Zhang G (2025) Path planning of oil spill recovery system with double USVs based on artificial potential field method. Journal of Marine Science and Application 24: 606-618. https://doi.org/10.1007/s42452-024-02237-8

Lin Q, Gou X, Tian P, Zhang H (2025) RL-Based USV path planning under the marine multimodal features considerations. IEEE Internet of Things Journal 12(11): 15274-15287. https://doi.org/10.1109/JIOT.2025.3530534

Luo J, Su Y (2025) Path planning for Multi-USV target coverage in complex ocean environments. Ocean Engineering 312: 119090. https://doi.org/10.1016/j.oceaneng.2024.119090

Mozzen H, Molaei S, Farzinavash L, Sabaei M (2023) PSO-ELPM: PSO with elite learning, enhanced parameter updating, and exponential mutation operator. Information Sciences 628: 70-91. https://doi.org/10.1016/j.ins.2023.01.103

Nguyen LV, Kwok NM, Ho QP (2024) Fermat-Weber location particle swarm optimization for cooperative path planning of unmanned aerial vehicles. Applied Soft Computing 167: 112269. https://doi.org/10.1016/j.asoc.2024.112269

Qiu H, Xue B, Xiao G, Niu B (2023) Q-learning based particle swarm optimization with multi-exemplar and elite learning. Lecture Notes in Computer Science 14086: 1-12. https://doi.org/10.1007/978-981-99-4755-3-27

Ratnaweera A, Halgamuge SK, Watson HC (2004) Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients. IEEE Transactions on Evolutionary Computation 8: 240-255. https://doi.org/10.1109/TEVC.2004.630731

Shen Y, Liu J, Luo Y (2021) Review of path planning algorithms for unmanned vehicles. 2021 IEEE 2nd International Conference on Information Technology, Big Data and Artificial Intelligence (ICIBAI), IEEE, Nanchang, China, 968-964. https://doi.org/10.1109/ICIBAI5261.2021.9688064

Sun BN (2024) Multi-AUVs cooperative path planning in 3D underwater terrain and vortex environments based on improved multi-objective particle swarm optimization algorithm. Ocean Engineering 311: 118944. https://doi.org/10.1016/j.oceaneng.2024.118944

Wang P, Liu R, Tian X, Zhang X, Qiao L, Wang Y (2023a) Obstacle avoidance for environmentally-driven USVs based on deep reinforcement learning in large-scale uncertain environments. Ocean Engineering 270: 113670. https://doi.org/10.1016/j.oceaneng.2023.113670

Wang Y, Wang B, Li Z, Xu C (2023b) A novel particle swarm optimization based on hybrid-learning model. Mathematical Biosciences and Engineering 20(4): 7056-7087. https://doi.org/10.3934/mbe.2023305

Wen N, Long Y, Zhang R, Liu G, Wan W, Jiao D (2023) COLREGs-based path planning for USVs using the deep reinforcement learning strategy. Journal of Marine Science and Engineering 11(12): 2334. https://doi.org/10.3390/jmse11122334

Westhedler J, Rickin J, Popovic M (2023) Multi-UAV adaptive path planning using deep reinforcement learning. 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, Detroit, USA, 1-5. https://doi.org/10.1109/IR055552.2023.10342516

Xiao H, Fu L, Shang C, Lin Y, Fan Y (2023) Multiobjective cooperative path planning of uncrewed surface vehicle based on deep reinforcement learning. IEEE Internet of Things Journal 12: 232. https://doi.org/10.1109/JIOT.2023.3280232

Yan M, Chan CA, Gygax AF, Li C, Nirmalathas A, Chih-Lin I (2024) Efficient generation of optimal UAV trajectories with uncertain obstacle avoidance in MEC networks. IEEE Internet of Things Journal 11(23): 38380-38392. https://doi.org/10.1109/JIOT.2024.3446664

Yang Q, Guo X, Gao XD, Xu DD, Ku DD, Lu ZY (2022) Differential elite learning particle swarm optimization for global numerical optimization. Mathematics 10: 1261. https://doi.org/10.3390/math10081261

Zare M, Akbari MA, Aizijanzadeh R, Barghoose M (2023) A modified particle swarm optimization algorithm with enhanced search quality and population using hummingbird flight patterns. Decision Analytics Journal 7: 100251. https://doi.org/10.1016/j.dajour.2023.100251

Zhang J, Li K (2024) Fusion of nonlinear inertia weight and probability mutation for binary particle swarm optimization algorithm. Communications in Computer and Information Science 2146: 39-48. https://doi.org/10.1007/978-981-97-4393-34

Zhang J, Ren J, Cui Y, Fu D, Cong J (2024) Multi-USV task planning method based on improved deep reinforcement learning. IEEE Internet of Things Journal 11(10): 18549-18567. https://doi.org/10.1109/JIOT.2024.3363044

Zhang X, Lin Q (2022) Three-learning strategy particle swarm algorithm for global optimization problems. Information Sciences 593: 289-313. https://doi.org/10.1016/j.ins.2022.01.075

Zhao Y, Ma Y, Hu S (2021) USV formation and path-following control via deep reinforcement learning with random braking. IEEE Transactions on Neural Networks and Learning Systems 32(12): 5468-5478. https://doi.org/10.1109/TNNLS.2021.3068762

Zhan ZH, Zhang J, Li Y, Chung HSH (2009) Adaptive particle swarm optimization. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 39(6): 1362-1381. https://doi.org/10.1109/TSMCB.2009.2015956

Zhou X, Zhou S, Han Y, Zhu S (2022) Lévy flight-based inverse adaptive comprehensive learning particle swarm optimization. Mathematical Biosciences and Engineering 19(5): 5241-5268. https://doi.org/10.3934/mbe.2022246

Zhou Y, Gong C, Chen K (2025) Adaptive control scheme for USV trajectory-tracking under complex environmental disturbances via deep reinforcement learning. IEEE Internet of Things Journal 12(11): 15181-15196. https://doi.org/10.1109/JIOT.2025.3528049