Journal of Marine Science and Application (2026) 25: 1471-1489
https://doi.org/10.1007/s11804-025-00789-z

RESEARCH ARTICLE

Energy-Efficient AUV Path Planning Under Multiple Constraints Using a Synergistic Differential Evolution Algorithm

Jiehui Tan[1], Yushan Sun[1], Kaiqian Cai[1,2], Yinghao Zhang[3] and Liwen Zhang[1]

Received: 25 April 2025 / Accepted: 13 October 2025
© Harbin Engineering University and Springer-Verlag GmbH Germany, part of Springer Nature 2026

Abstract

Path planning is essential for an autonomous underwater vehicle (AUV) to efficiently perform underwater missions. To address the challenges of energy-efficient three-dimensional path planning under multiple constraints, this study proposes a synergistic differential evolution (SDE) algorithm. First, a comprehensive underwater environment model is developed, including a path encoding model and a deterministic path-evaluation method. Second, a synergistic evolution mechanism is introduced to improve global exploration by adaptively sharing the evolutionary information of two synergistic populations. Third, evolutionary guidance is proposed to refine the search direction by integrating successful population movements based on fitness improvements and positional correlations. Finally, a cosine-based strategy is employed to adjust the population size by eliminating low-quality solutions. Compared to state-of-the-art methods, the proposed algorithm achieves average improvements of 2.37% in best fitness, 15.99% in mean fitness, 13.42% in median fitness, 80.69% in standard deviation, 66.65% in convergence quality, and 39.91% in runtime. In addition, AUV field tests further validate the feasibility and reliability of the SDE-based path-planning approach.

Keywords Autonomous underwater vehicle; Path planning; Energy optimization; Differential evolution; Swarm intelligence

1 Introduction

The increasing demand for underwater exploration and resource use has driven the need for more efficient and reliable underwater operations. Autonomous underwater vehicles (AUVs) have gained broad application due to their ease of deployment, flexible sensor configurations, and high environmental adaptability. These advantages have made AUVs increasingly popular for applications such as data collection (Yan and Li, 2023), sea ranching (Hu et al., 2023), tunnel inspection (Sun et al., 2025) and water-quality monitoring (Chang et al., 2021). However, the limited onboard energy poses significant challenges and requires effective strategies for energy optimization while ensuring safe and reliable operation.

Path planning is a critical component of AUV autonomy, guiding it to its destination based on an optimal criterion while satisfying multiple constraints. Energy-efficient navigation is crucial given the limited onboard energy supply, particularly in the presence of ocean currents that significantly impact energy consumption. Additionally, ensuring safe and stable operation must consider spatial constraints imposed by terrain and obstacles (Zhai et al., 2024), as well as attitude constraints that affect the AUV's maneuverability and onboard sensor functionality. These constraints challenge the optimality, algorithmic stability, and computational efficiency, directly influencing mission efficiency (Li et al., 2024). Therefore, developing a high-performance path-planning algorithm that effectively addresses multiple constraints while optimizing energy efficiency is critical.

Swarm intelligence optimization is a current research hotspot for energy optimization and constraint handling in

Article Highlights

- A synergistic differential evolution algorithm is proposed for energy-efficient AUV path planning under multiple constraints in complex 3D environments.
- A synergistic evolution mechanism is introduced to enhance global exploration, improve optimization quality, and reduce the risk of local optimum entrapment.
- A weighted evolutionary guidance strategy refines the search direction, enhancing algorithmic stability and convergence.
- A cosine-based population-size adjustment strategy alleviates computational burden while maintaining optimization performance.

✉ Yushan Sun
sunyushan@hrbeu.edu.cn

[1] National Key Laboratory of Autonomous Marine Vehicle Technology, Harbin Engineering University, Harbin 150001, China

[2] Nanhai Institute of Harbin Engineering University, Sanya 572025, China

[3] Wuhan Second Ship Design and Research Institute, Wuhan 430200, China

AUV path planning (Cheng et al., 2021). An enhanced water wave optimization method based on the elite opposition-based learning strategy and the simplex method was proposed to solve the shortest path problem in an obstacle environment (Yan et al., 2021). Tentative crossover, path self-smoothing, and adaptive adjustment of genetic operation probabilities were proposed to address the three-dimensional (3D) gridded AUV path-planning problem (Wen et al., 2021). A success-history-based adaptive differential evolution was proposed to optimize the diving trajectory with motion constraints (Chen et al., 2021). An adaptive genetic algorithm for path generation and a collision detection mechanism with a path-recovery strategy for global path adjustment were proposed (Hao et al., 2022). An improved particle swarm optimization (PSO) algorithm was proposed to optimize the energy cost and address the maneuverability constraints (Zhang et al., 2022). However, the stability and computational efficiency of the proposed algorithm have not been fully verified. A whale optimization algorithm based on segment learning and adaptive operator selection were proposed to solve the path-planning problem under the influence of natural terrain and ocean currents (Huang et al., 2023). An improved PSO algorithm was proposed to alleviate premature convergence and local optimality (Li and Yu, 2023). A hybrid gray wolf optimization algorithm was proposed to optimize both distance and collision penalties (Sahoo et al., 2023). An enhanced A* algorithm as an initializer and a genetic algorithm were proposed to address the constrained path-planning problem (Chen et al., 2024). An improved artificial bee colony algorithm was used to guide the AUV through an obstacle environment with vortices (Xu et al., 2024), ensuring it reaches the target at an optimal angle. A double-layer coding model was used to remove invalid path points and an improved differential evolution algorithm was proposed to optimize travel distance and various constraint penalties (Fan and Qu, 2024). However, its algorithmic stability decreases in more complex environments, as indicated by a higher standard deviation. An arithmetic optimization algorithm was used to guide global search and the sparrow search algorithm was applied to focus on local exploration, while a cross-integration mutation strategy was established to address AUV path planning with multiple constraints in a single-vortex environment (Liu et al., 2024). Due to the integration of two distinct algorithms, it requires tuning several parameters and may become unstable in increasingly complex environments.

As explained in Table 1, AUV path planning has become increasingly complex, evolving from two-dimensional (2D) to 3D spaces, from distance-based optimization to energy-efficient navigation in flow fields, and from single-constraint to multiple constraint problems. However, research that simultaneously addresses energy optimization and multiple constraints in complex flow-field environments remains underdeveloped. This is due to four main challenges (Peng et al., 2023, Rahimi et al., 2022):

Table 1 Recent developments in autonomous underwater vehicle (AUV) path planning. √ indicates that the cited work considers the corresponding constraint; × indicates it does not

References	Dim	Flow	Obstacle	Terrain	Pitch	Yaw
Yan et al. (2021)	2D	×	√	×	×	×
Wen et al. (2021)	3D	√	√	×	×	×
Chen et al. (2021)	3D	√	×	×	√	√
Hao et al. (2022)	3D	√	√	√	×	×
Zhang et al. (2022)	3D	√	×	√	√	√
Huang et al. (2023)	2D	√	×	√	×	×
Li and Yu (2023)	3D	√	×	√	×	×
Sahoo et al. (2023)	3D	×	√	×	×	×
Chen et al. (2024)	2D	√	√	√	×	√
Xu et al. (2024)	2D	√	√	×	×	√
Fan and Qu (2024)	3D	×	√	√	√	√
Liu et al. (2024)	3D	√	√	√	×	×

1) Multiple local minima caused by complex flow environments;

2) Spatial constraints imposed by the terrain and obstacles reduce the available free space and increase the solution space discontinuity, while attitude constraints introduce additional coupling between state variables, further hindering global exploration;

3) Increasing environmental complexity and constraint diversity undermine algorithmic stability; and

4) The presence of multiple constraints generates numerous low-quality solutions, reducing computational efficiency.

To address these challenges, this work proposes a synergistic differential evolution (SDE) algorithm for energy-efficient AUV path planning under multiple constraints in complex 3D environments. The main contributions are summarized as follows:

1) A comprehensive underwater environment model is constructed, including a 3D path encoding model and a deterministic fitness evaluation mechanism.

2) A synergistic evolution mechanism is proposed to improve global exploration by adaptively sharing the evolutionary information of two synergistic populations, which improves the optimization and reduces the risk of entrapment in local optima.

3) A weighted evolutionary guidance is proposed to improve the search direction by integrating successful movements based on fitness improvement and positional correlation, thereby enhancing algorithmic stability and convergence.

4) A cosine-based strategy is proposed to dynamically adjust the population size by effectively eliminating low-quality solutions, thus alleviating the synergistic evolution computational burden while preserving the optimization.

The remainder of this work is structured as follows: Sec-

tion 2 presents the AUV path-planning problem. Section 3 details the proposed SDE algorithm. Section 4 contains analysis and discussion of the experimental results, while Section 5 provides the research summary and future outlook.

2 Problem statement

2.1 Coordinate systems

As illustrated in Figure 1, the Earth-fixed coordinate system employed in this study is the North East Down coordinate system (Prestero, 2001), where the X-axis points north, the Y-axis east, and the Z-axis downward, perpendicular to the XOY plane. Additionally, a body-fixed coordinate system is coupled to the AUV, with the X_B-axis pointing forward, the Y_B-axis to the right, and the Z_B-axis downward.

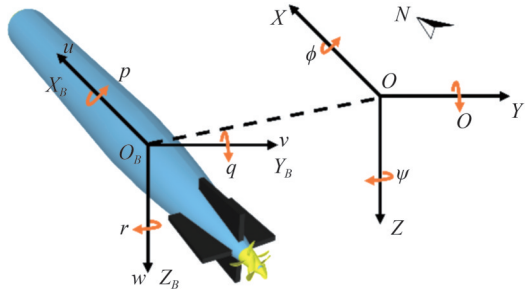


Figure 1 Coordinate system

2.2 Flow-field model

Based on the Lamb–Oseen vortex model (Fang et al., 2022; Garau et al., 2006), the flow field velocity (c_x, c_y) is defined in Eq. (1),

$$\begin{cases} c_x(x,y) = \sum_{i=1}^{n^v} \Gamma_i \dfrac{y - y_i^v}{2\pi\rho_i^2}\left(1 - \exp\left(-\rho_i^2/\delta_i^2\right)\right) \\ c_y(x,y) = \sum_{i=1}^{n^v} -\Gamma_i \dfrac{x - x_i^v}{2\pi\rho_i^2}\left(1 - \exp\left(-\rho_i^2/\delta_i^2\right)\right) \end{cases} \tag{1}$$

where $\rho_i = \left\|(x - x_i^v, y - y_i^v)\right\|_2$. (x_i^v, y_i^v), Γ_i, and δ_i denote the position, strength, and radius of the ith vortex, respectively, and n^v is the number of the vortices.

2.3 Terrain model

Based on an exponential function (Qi et al., 2010, Sun and Niu, 2024), the terrain is modeled as given in Eq. (2).

$$Z_i(x,y) = z_{\max} - \sum_{i=1}^{n^h} h_i \exp\left[-\frac{\left(x - x_i^h\right)^2}{\left(x_i^a\right)^2} - \frac{\left(y - y_i^h\right)^2}{\left(y_i^a\right)^2} \right] \tag{2}$$

where $z_{\max}$ is the maximum depth, n^h is the number of hills, h_i represents the height of the ith hill, (x_i^h, y_i^h) denotes its center position, and (x_i^a, y_i^a) controls the intensity of the height attenuation.

2.4 Path encoding model

As illustrated in Figure 2, the path from the start $\boldsymbol{P}_S$ to the goal $\boldsymbol{P}_G$ is defined by a series of potential waypoints $\boldsymbol{P}_i$, each characterized by a polar angle and a depth value (φ_i, z_i),

$$\boldsymbol{P}_i = (x_i, y_i, z_i) = \left(i \cdot r_c \cos(\varphi_i) + x_s, i \cdot r_c \sin(\varphi_i) + y_s, z_i\right) \tag{3}$$

where $r_c = \left\|(x_G - x_S, y_G - y_S)\right\|_2 / (n+1)$. As shown in Figure 2, the polar angle φ rotates around the plumbline of $\boldsymbol{P}_S$, conforming to the right-hand rule, and points north when $\varphi = 0$. Considering the general existence of the space boundary, it is necessary to limit the range of φ_i $[\varphi_{\min,i}, \varphi_{\max,i}]$, as detailed in Appendix A.

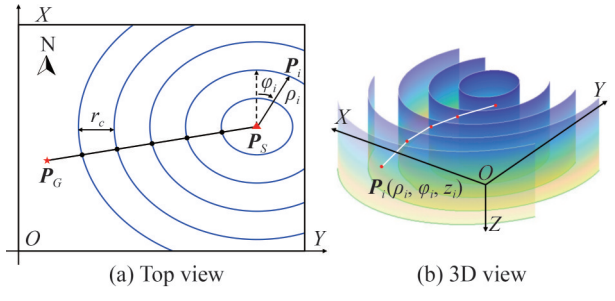


Figure 2 Path encoding

To reduce discontinuities, cubic spline interpolation is applied to insert m points $\boldsymbol{P}_{i,j}$ $(j = 1, 2, \cdots, m$, $\boldsymbol{P}_{i,0} = \boldsymbol{P}_i$, $\boldsymbol{P}_{i,m+1} = \boldsymbol{P}_{i+1})$ into $\boldsymbol{P}_i\boldsymbol{P}_{i+1}$ uniformly.

$$\begin{cases} x_i(r) = a_{x,i} + b_{x,i}(r - r_i) + c_{x,i}(r - r_i)^2 + d_{x,i}(r - r_i)^3 \\ y_i(r) = a_{y,i} + b_{y,i}(r - r_i) + c_{y,i}(r - r_i)^2 + d_{y,i}(r - r_i)^3 \\ z_i(r) = a_{z,i} + b_{z,i}(r - r_i) + c_{z,i}(r - r_i)^2 + d_{z,i}(r - r_i)^3 \end{cases} \tag{4}$$

where $r = r_i + (j \cdot r_c)/(m+1)$ and $r_i = i \cdot r_c$. a, b, c, and d are the coefficients to be solved.

2.5 Path evaluation

To ensure mission energy efficiency and safety, this work focuses on minimizing the energy cost J_e while handling various constraints, including restricted region J_r, terrain J_t, pitch angle J_p, and yaw change J_y. The overall fitness function is defined as follows:

$$\min J = \omega_1 J_e + \omega_2 J_r + \omega_3 J_t + \omega_4 J_p + \omega_5 J_y \quad (5)$$

where ω is the weight coefficient of the fitness function.

2.5.1 Energy consumption cost

The absolute velocity of the AUV is the linear superposition of its velocity relative to the surrounding flow field and the flow-field velocity itself. When the AUV moves along the planned path with a constant relative surge speed u_{const}, while counteracting the flow field in the AUV's lateral and vertical directions, its velocity **RV** relative to the water flow in the body-fixed coordinate system is given by:

$$\begin{bmatrix} \mathrm{RV}_x \\ \mathrm{RV}_y \\ \mathrm{RV}_z \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ \sin(\psi) & -\cos(\psi) & 0 \\ -\cos(\psi)\sin(\theta) & -\sin(\psi)\sin(\theta) & 0 \end{bmatrix} \begin{bmatrix} c_x \\ c_y \\ 0 \end{bmatrix} + \begin{bmatrix} u_{\text{const}} \\ 0 \\ 0 \end{bmatrix} \quad (6)$$

where ψ is the yaw angle and θ is the pitch angle, assuming the roll angle is 0°.

Hydrodynamic drag is the main cause of energy loss, and its energy consumption power is proportional to the cube of **RV** (Luo et al., 2024; Xu et al., 2024). Therefore, energy cost J_e can be calculated as follows:

$$J_e = \sum_{i=0}^{n} \sum_{j=0}^{m} \sum_{k}^{x,y,z} \alpha_k \left| \overline{\mathrm{RV}}_{k,i,j} \right|^3 t_{i,j} \quad (7)$$

$$\overline{\mathrm{RV}}_{k,i,j} = 0.5\left(\mathrm{RV}_k\left(\boldsymbol{P}_{i,j} \right) + \mathrm{RV}_k\left(\boldsymbol{P}_{i,j+1} \right) \right) \quad (8)$$

$$t_{i,j} = 2\left| \boldsymbol{P}_{i,j} \boldsymbol{P}_{i,j+1} \right| \Big/ \left(\mathrm{EV}\left(\boldsymbol{P}_{i,j} \right) + \mathrm{EV}\left(\boldsymbol{P}_{i,j+1} \right) \right) \quad (9)$$

$$\mathrm{EV} = u_{\text{const}} + c_x \cos(\psi)\cos(\theta) + c_y \sin(\psi)\cos(\theta) \quad (10)$$

where α_k is a drag coefficient, $\overline{\mathrm{RV}}_{k,i,j}$ is the average relative velocity, EV is the AUV speed along the path in the Earth-fixed coordinate system, and $\boldsymbol{P}_{i,j}$ is a cubic spline interpolation point of $\boldsymbol{P}_i \boldsymbol{P}_{i+1}$.

2.5.2 Restricted region constraint cost

To prevent planned paths from intruding into restricted zones, such as offshore infrastructure areas or underwater operation regions, a penalty is applied that increases as any path point $\boldsymbol{P}_{i,j}$ approaches the center of the restricted region. The corresponding constraint cost is calculated as follows:

$$J_r = \sum_{i=0}^{n} \sum_{j=0}^{m} \sum_{k=1}^{n^r} R_k\left(\boldsymbol{P}_{i,j} \right) \quad (11)$$

$$R_k\left(\boldsymbol{P}_{i,j} \right) = \begin{cases} r_{R,k} - \left| \boldsymbol{P}_{R,k} \boldsymbol{P}_{i,j} \right| & \text{if } r_{R,j} \geqslant \left| \boldsymbol{P}_{R,k} \boldsymbol{P}_{i,j} \right| \\ 0 & \text{otherwise} \end{cases} \quad (12)$$

where n^r denotes the number of restricted regions. $\boldsymbol{P}_{R,k}$ and $r_{R,k}$ represent the horizontal center position and radius of the kth circular restricted region, respectively.

2.5.3 Terrain constraint cost

To ensure safe seabed clearance and prevent collisions caused by terrain undulations, a terrain constraint is imposed. Specifically, if any path point $\boldsymbol{P}_{i,j}$ exceeds a predefined depth threshold, a penalty is imposed to discourage unsafe descent. The terrain constraint cost is calculated as follows:

$$J_t = \sum_{i=0}^{n} \sum_{j=0}^{m} H\left(\boldsymbol{P}_{i,j} \right) \quad (13)$$

$$H\left(\boldsymbol{P}_{i,j} \right) = \max\left(z_{i,j} - Z_h\left(x_{i,j}, y_{i,j} \right) + d_{\text{safe}}, 0 \right) \quad (14)$$

where d_{safe} represents the safety redundancy for depth, and $Z_h\left(x_{i,j}, y_{i,j} \right)$ is the terrain depth at point $\boldsymbol{P}_{i,j}$.

2.5.4 Pitch angle constraint cost

Excessive pitch angle variations can destabilize heave motion and impair the performance of onboard sensors, such as the Doppler Velocity Log, which operates with a limited line of sight. To promote smoother depth transitions and enhance navigational stability, the pitch angle constraint cost is defined as follows:

$$J_p = \sum_{i=0}^{n} \sum_{j=0}^{m} A_p\left(\theta_{i,j} \right) \quad (15)$$

$$A_p\left(\theta_{i,j} \right) = \begin{cases} 50 & \left| \theta_{i,j} \right| > \pi/18 \\ 0 & \text{otherwise} \end{cases} \quad (16)$$

$$\theta_{i,j} = \arcsin\left(\left(z_{i,j+1} - z_{i,j} \right) \Big/ \left| \boldsymbol{P}_{i,j} \boldsymbol{P}_{i,j+1} \right| \right) \quad (17)$$

2.5.5 Yaw angle constraint cost

Excessive changes in yaw angle can compromise the AUV's navigational stability and result in infeasible turns. To ensure the feasibility and smoothness of horizontal maneuvers, a yaw angle constraint cost is formulated as:

$$J_y = \sum_{i=0}^{n} \sum_{j=0}^{m-1} A_y\left(\Delta\psi_{i,j} \right) \quad (18)$$

$$A_y\left(\Delta\psi_{i,j} \right) = \begin{cases} 50 & \left| \psi_{i,j+1} - \psi_{i,j} \right| > \pi/4 \\ 0 & \text{otherwise} \end{cases} \quad (19)$$

$$\psi_{i,j} = \mathrm{atan2}\left(y_{i,j+1} - y_{i,j}, x_{i,j+1} - x_{i,j} \right) \quad (20)$$

where atan2() refers to the standard two argument arctangent function.

3 Proposed method

3.1 AUV path planning based on the SDE algorithm

Traditional methods employ a single population to address constrained optimization problems, guided by a unified cost function that combines the main objective with penalty functions. However, AUV path planning involves multiple safety and feasibility constraints that may take precedence over energy optimization. In such cases, single-population approaches are insufficient to fully explore the solution space while balancing the constraints and the main objective. Moreover, the presence of constraints not only increases the solution space complexity but also heightens the risk of local optima.

Reflecting the above limitations, the main population $\boldsymbol{P}_{\text{main}}$ and the auxiliary population $\boldsymbol{P}_{\text{aid}}$ are proposed. In the following selection operation, $\boldsymbol{P}_{\text{main}}$ is guided by the overall fitness function in Eq. (5), while $\boldsymbol{P}_{\text{aid}}$ is evaluated only by the energy cost without constraints given in Eq. (7). In this case, $\boldsymbol{P}_{\text{main}}$ tends to search within the feasible region, while $\boldsymbol{P}_{\text{aid}}$ can explore the entire solution space without constraints.

The framework of the SDE algorithm is shown in Figure 3. It comprises an adaptive synergistic evolution mechanism, a mutation strategy incorporating weighted evolutionary guidance, and a cosine based population size adjustment strategy.

3.1.1 Initialization

The individual $\boldsymbol{x}_{i,0}$ is initialized as follows:

$$x_{i,0}^{j} = x_{\min}^{j} + \text{rand} \times \left(x_{\max}^{j} - x_{\min}^{j}\right) \tag{21}$$

where $i = 1, 2, \cdots, \text{NP}$ and $j = 1, 2, \cdots, D$. NP is the population size, and rand is a random number in (0, 1). D, $\boldsymbol{x}_{\max}$ and $\boldsymbol{x}_{\min}$ denote the dimension and the boundary, respectively.

3.1.2 Mutation with weighted evolutionary guidance

The traditional mutation operator relies on the guidance of the current optimal individuals and a random differential vector (Zhang and Sanderson, 2009). Although the introduction of the random differential vector enhances population diversity, it also introduces uncertainty in the evolutionary direction. Under multiple constraint conditions, this uncertainty increases the likelihood of generating infeasible solutions, thereby reducing both stability and convergence.

Guided by the fitness function, individuals $\boldsymbol{x}$ exhibit positional correlation and similarity during iterations. Leveraging the successful movements of related individuals can enhance the search direction. To achieve this, a weighted evolutionary guidance term $\boldsymbol{x}^{w}$ is incorporated into the mutation operator in Eq. (22). As shown in Eqs. (23)–(24), the term $\boldsymbol{x}^{w}$ records the weighted movement of individuals that successfully improve their fitness, considering both the magnitude of fitness improvement and the positional similarity. This reduces random fluctuations and propagates the global optimization trend, thereby enhancing the algorithmic stability and its convergence efficiency.

$$\boldsymbol{v}_{i,g} = \boldsymbol{x}_{i,g} + F_i \cdot \left(\boldsymbol{x}_{\text{pbest},g} - \boldsymbol{x}_{i,g}\right) + F_i \cdot \left(\boldsymbol{x}_{r1,g} - \boldsymbol{x}_{r2,g}\right) + F_i \cdot \boldsymbol{x}_{i,g-1}^{w} \tag{22}$$

$$\boldsymbol{x}_{i,g}^{w} = \frac{\sum_{j \neq i}^{|\boldsymbol{u}_g^s|} \left(\boldsymbol{u}_{j,g} - \boldsymbol{x}_{j,g}\right) \cos\left(\boldsymbol{x}_{j,g}, \boldsymbol{x}_{i,g+1}\right) \left(J\left(\boldsymbol{u}_{j,g}\right) - J\left(\boldsymbol{x}_{j,g}\right)\right)}{\sum_{k \neq i}^{|\boldsymbol{u}_g^s|} \left(J\left(\boldsymbol{u}_{k,g}\right) - J\left(\boldsymbol{x}_{k,g}\right)\right)} \tag{23}$$

$$\cos\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right) = \frac{\boldsymbol{x}_i' \cdot \boldsymbol{x}_j'}{\left\|\boldsymbol{x}_i'\right\| \cdot \left\|\boldsymbol{x}_j'\right\|}, \boldsymbol{x}' = \frac{\boldsymbol{x} - \boldsymbol{x}_{\min}}{\boldsymbol{x}_{\max} - \boldsymbol{x}_{\min}} \tag{24}$$

where $\boldsymbol{v}_{i,g}$ is a variation vector, F_i is an adaptive mutation factor (Tanabe and Fukunaga, 2013), $\boldsymbol{x}_{\text{pbest},g}$ is randomly selected from the top p% of individuals based on fitness ranking, and $\boldsymbol{x}_{r1,g}$ and $\boldsymbol{x}_{r2,g}$ are randomly selected individuals. $\boldsymbol{u}_g^s$ represents a set of trial vectors that succeed in the selection operation. $\boldsymbol{x}_{i,0}^{w} = 0$.

If $\boldsymbol{v}_{i,g}$ is out of the boundary, it is corrected as follows:

$$v_{i,g}^{j} = \begin{cases} \left(x_{\min}^{j} + x_{i,g}^{j}\right)/2 & \text{if } v_{i,g}^{j} < x_{\min}^{j} \\ \left(x_{\max}^{j} + x_{i,g}^{j}\right)/2 & \text{if } v_{i,g}^{j} > x_{\max}^{j} \end{cases} \tag{25}$$

3.1.3 Binomial crossover

The trial vector $\boldsymbol{u}_{i,g}$ is generated by applying binomial crossover as follows:

$$u_{i,g}^{j} = \begin{cases} v_{i,g}^{j} & \text{if rand} \leqslant \text{CR}_i \text{ or } j = j_r \\ x_{i,g}^{j} & \text{otherwise} \end{cases} \tag{26}$$

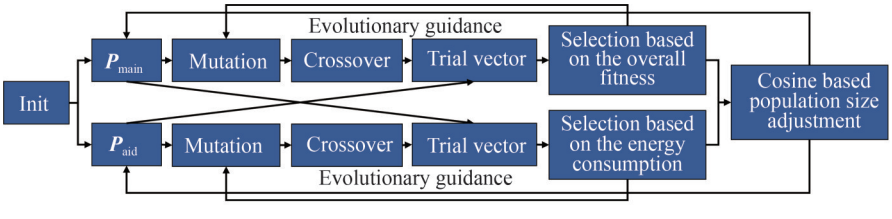


Figure 3 Basic framework of the proposed algorithm

where CR_i is an adaptive crossover factor (Tanabe and Fukunaga, 2013) and j_r is a random integer in [1, D].

3.1.4 Adaptive synergistic evolution mechanism

Before the selection operation, it is proposed to exchange the trial vector $\boldsymbol{u}_{i,g}$ adaptively between $\boldsymbol{P}_{\text{main}}$ and $\boldsymbol{P}_{\text{aid}}$. Because $\boldsymbol{P}_{\text{main}}$ considers constraints during the energy optimization process, it can guide $\boldsymbol{P}_{\text{aid}}$ to explore nearby infeasible regions by providing feasible solution references. In return, as $\boldsymbol{P}_{\text{aid}}$ is not restricted by constraints during optimization, the promising solutions it discovers enhance the population diversity and improve the global search capability of $\boldsymbol{P}_{\text{main}}$.

$$\boldsymbol{u}_{i,g} = \begin{cases} \boldsymbol{x}'_{i_r,g} & \text{if rand} \leqslant \text{ER}_i \\ \boldsymbol{u}_{i,g} & \text{otherwise} \end{cases} \tag{27}$$

where ER_i is an exchange probability and $\boldsymbol{x}'_{i_r,g}$ is a random individual from another population.

Considering the effectiveness of exchanged information during the iteration, an adaptive reference pool (**ARP**) is proposed to dynamically guide the adjustment of **ER**.

$$\text{ER}_i = \begin{cases} 0 & \text{if } \text{ARP}_{i_r} = \perp \\ \text{randn}\left(\text{ARP}_{i_r}, 0.1\right) & \text{otherwise} \end{cases} \tag{28}$$

where $\perp$ is a terminator value, randn() generates a normally distributed random number, APR_{i_r} is a random element of the **ARP**. The elements of the **ARP** are all initialized to 0.5. If all of the elements of $\mathbf{ARP}_{\text{main}}$ are set to $\perp$, the population exchange and evolution of $\mathbf{P}_{\text{aid}}$ will stop for computational efficiency. Otherwise, the **ARP** is updated as follows:

$$\mathbf{ARP}_{k,g+1} = \begin{cases} \perp & \text{if } \boldsymbol{E}_u \neq \varnothing \text{ and } \boldsymbol{S}_{\text{ER}} = \varnothing \\ \text{mean}\left(\boldsymbol{S}_{\text{ER}}\right) & \text{if } \boldsymbol{S}_{\text{ER}} \neq \varnothing \\ \mathbf{ARP}_{k,g} & \text{otherwise} \end{cases} \tag{29}$$

where $\boldsymbol{E}_u$ and $\boldsymbol{S}_{\text{ER}}$ are empty sets at the start of each iteration, $\boldsymbol{E}_u$ records the exchanged trial vector $\boldsymbol{u}_{i,g}$; $\boldsymbol{S}_{ER}$ records the ER_i corresponding to the exchanged $\boldsymbol{u}_{i,g}$ that succeeds in the selection. The function mean() calculates the average value.

3.1.5 Selection

The individuals in the next generation are generated as follows:

$$\boldsymbol{x}_{i,g+1} = \begin{cases} \boldsymbol{u}_{i,g} & \text{if } J\left(\boldsymbol{u}_{i,g}\right) < J\left(\boldsymbol{x}_{i,g}\right) \\ \boldsymbol{x}_{i,g} & \text{otherwise} \end{cases} \tag{30}$$

3.1.6 Cosine-based population-size adjustment

The presence of multiple constraints may result in a large number of infeasible solutions, which not only degrade the optimization efficiency but also place additional strain on limited computational resources. To address this problem, a cosine based population size adjustment strategy is proposed as shown in Eqs. (31) – (32). If $\text{NP}_{g+1} < \text{NP}_g$, the worst-ranked $\left(\text{NP}_g - \text{NP}_{g+1}\right)$ individuals will be eliminated.

$$\text{NP}_g = \text{round}\left(\text{NP}^{\text{init}} \cdot \text{PPC}_g\right) \tag{31}$$

$$\text{PPC}_g = \left(1 - \text{PPC}_G\right) \cdot \cos^2\left(0.5\pi\left(g/G\right)^{\lambda}\right) + \text{PPC}_G \tag{32}$$

where PPC_g denotes the percentage of the population size at generation g. The function round() is the rounding operator. G is the maximum number of generations, and NP^{init} is the initial population size. The hyperparameter $\lambda > 0$ controls the rate of reduction. A larger value of λ results in a slower reduction of PPC_g, as shown in Figure 4.

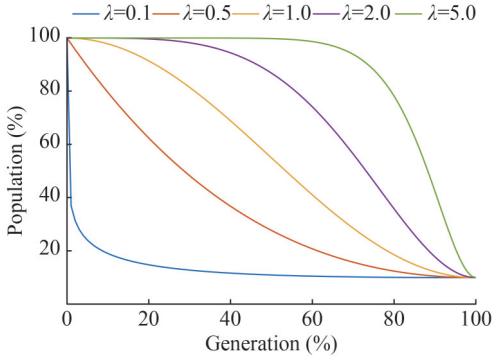


Figure 4 Population-size adjustment curves

3.2 Algorithmic analysis

To avoid unnecessary repetition, the framework of the SDE-based AUV path-planning pseudocode is presented in Algorithm 1, while the evolutionary function for $\boldsymbol{P}_{\text{main}}$ and $\boldsymbol{P}_{\text{aid}}$ is shown in Algorithm 2.

3.2.1 Time complexity

The approximate time complexity of the evolution for $\boldsymbol{P}_{\text{main}}$ is

$$\text{O}\left(D \cdot \text{NP}^{\text{init}} \int_0^G \text{PPC}_g \text{d}g\right) = \text{O}\left(0.5D \cdot \text{NP}^{\text{init}} G\left(1 + \text{PPC}_G + \left(1 - \text{PPC}_G\right)\int_0^1 \cos\left(\pi x^{\lambda}\right)\text{d}x\right)\right) \tag{33}$$

because $-1 < \cos\left(\pi x^{\lambda}\right) < 1$, then

$$\begin{aligned} &\text{O}\left(D \cdot \text{NP}^{\text{init}} \int_0^G \text{PPC}_g \text{d}g\right) \geqslant \text{O}\left(D \cdot \text{NP}^{\text{init}} \cdot G \cdot \text{PPC}_G\right) \\ &\text{O}\left(D \cdot \text{NP}^{\text{init}} \int_0^G \text{PPC}_g \text{d}g\right) \leqslant \text{O}\left(D \cdot \text{NP}^{\text{init}} \cdot G\right) \end{aligned} \tag{34}$$

Algorithm 1 AUV path-planning algorithm based on SDE

Initialize fitness functions, define start point $\boldsymbol{P}_S$ and goal point $\boldsymbol{P}_G$, calculate the boundary of the polar angles φ;
Define G; $g \leftarrow 0$, $\mathrm{NP}_g \leftarrow \mathrm{NP}^{\mathrm{init}}$;
Randomly initialize $\boldsymbol{P}_{\mathrm{main},g}$ and $\boldsymbol{P}_{\mathrm{aid},g}$;
Evaluate $\boldsymbol{P}_{\mathrm{main},g}$ and $\boldsymbol{P}_{\mathrm{aid},g}$ with Eq. (5) and Eq. (7), respectively;
Initialize $\mathbf{ARP}_{\mathrm{main}}$ and $\mathbf{ARP}_{\mathrm{aid}}$;
Set flag for whether or not $\boldsymbol{P}_{\mathrm{aid}}$ to evolve $\mathrm{Evol}_{\mathrm{aid}} \leftarrow$true;
while $g <= G$ do
  Input $\mathbf{ARP}_{\mathrm{main},g}$, $\boldsymbol{P}_{\mathrm{main},g}$, $\boldsymbol{P}_{\mathrm{aid},g}$ and NP_g to Algorithm 2 and get upgraded with the output;
  if $\mathrm{Evol}_{\mathrm{aid}}$ = true then
    Input $\mathbf{ARP}_{\mathrm{aid},g}$, $\boldsymbol{P}_{\mathrm{aid},g}$, $\boldsymbol{P}_{\mathrm{main},g}$ and NP_g to Algorithm 2 and get upgraded with the output;
  end if
  Calculate NP_{g+1} based on Eq. (31);
  if $\mathrm{NP}_{g+1} < \mathrm{NP}_g$ do
    Delete the worst-ranked $\mathrm{NP}_g - \mathrm{NP}_{g+1}$ individuals of $\boldsymbol{P}_{\mathrm{main},g+1}$ and $\boldsymbol{P}_{\mathrm{aid},g+1}$;
  end if
  $g \leftarrow g + 1$;
end while

Algorithm 2 Evolutionary function

input: $\mathbf{ARP}_g$, $\boldsymbol{P}_g$, P'_g, NP_g
output: $\mathbf{ARP}_{g+1}$, $\boldsymbol{P}_{g+1}$
$\boldsymbol{E}_u$, $\boldsymbol{S}_{\mathrm{ER}}$, $u_g^s \leftarrow \varnothing$;
if $\boldsymbol{P}_g = \boldsymbol{P}_{\mathrm{main}}$ then
  Define the fitness function f_{cost} based on Eq. (5);
else
  Define the fitness function f_{cost} based on Eq. (7);
end if
for i = 1 to NP_g do
  Calculate $\boldsymbol{v}_{i,g}$ based on the mutation operator with weighted evolutionary guidance;
  Calculate $\boldsymbol{u}_{i,g}$ with Eq. (26);
  Calculate ER_i based on Eq. (28);
  if rand < ER_i then
    Randomly select an individual from P'_g to replace $\boldsymbol{u}_{i,g}$; $\boldsymbol{E}_u \leftarrow \boldsymbol{E}_u \cup \boldsymbol{u}_{i,g}$;
  end if
  if $f_{\mathrm{cost}}(\boldsymbol{u}_{i,g}) < f_{\mathrm{cost}}(\boldsymbol{x}_{i,g})$ then
    $\boldsymbol{x}_{i,g+1} \leftarrow \boldsymbol{u}_{i,g}$, $u_g^s \leftarrow u_g^s \cup \boldsymbol{u}_{i,g}$;
    if $\boldsymbol{u}_{i,g}$ was exchanged, then
      $\boldsymbol{S}_{\mathrm{ER}} \leftarrow \boldsymbol{S}_{\mathrm{ER}} \cup \mathrm{ER}_i$;
    end if
  else
    $\boldsymbol{x}_{i,g+1} \leftarrow \boldsymbol{x}_{i,g}$;
  end if
end for
if $k <$ size of ($\mathbf{ARP}_g$) then k++; else $k \leftarrow 0$; end if
Update $\mathbf{ARP}_{k,g+1}$ based on Eq. (29);
if all elements of $\mathbf{ARP}_{\mathrm{main}} = \perp$ then $\mathrm{Evol}_{\mathrm{aid}} \leftarrow$false; end if
return $\mathbf{ARP}_{g+1}$, $\boldsymbol{P}_{g+1}$

Therefore, the approximate time complexity of $\boldsymbol{P}_{\mathrm{main}}$ is $O(D \cdot \mathrm{NP}^{\mathrm{init}} \cdot G)$ according to Big O notation. The complexity of $\boldsymbol{P}_{\mathrm{aid}}$ depends on the point at which it stops evolving. In the worst case, it has a complexity of $O(D \cdot \mathrm{NP}^{\mathrm{init}} \cdot G)$. In the best case, it has a complexity of $O(D \cdot \mathrm{NP}^{\mathrm{init}} \cdot |\mathbf{ARP}|)$ if $\boldsymbol{P}_{\mathrm{aid}}$ stops evolving in the $|\mathbf{ARP}|$th generation. Overall, the time complexity of SDE is $O(D \cdot \mathrm{NP}^{\mathrm{init}} \cdot G)$.

3.2.2 Space complexity

The space complexity of SDE results from the two populations. Initially, the maximum space complexity is $O(2D \cdot \mathrm{NP}^{\mathrm{init}})$. If the $\boldsymbol{P}_{\mathrm{aid}}$ ceases evolution and is discarded early, the minimum space complexity in the final generation reduces to $O(D \cdot \mathrm{NP}^{\mathrm{init}} \cdot \mathrm{PPC}_G)$. Therefore, SDE has an approximate space complexity of $O(D \cdot \mathrm{NP}^{\mathrm{init}})$ according to Big O notation.

3.2.3 Convergence analysis

Let the best fitness value for each generation be:

$$J_g = J\left(\boldsymbol{x}_{\mathrm{best},g}\right) \tag{35}$$

As shown in Eq. (30), because the individual only accepts the better trial vector to update its current solution, then

$$J_{g+1} \leqslant J_g \tag{36}$$

Meanwhile, as demonstrated from Eq. (5) to Eq. (20), it is evident that $J(x) \geqslant 0$. Hence, by the Monotone Convergence Theorem, the best fitness value is guaranteed to converge:

$$\lim_{g \to \infty} J_g = J^* \tag{37}$$

However, this convergent value may not represent the global optimum. Hence, statistical experiments are essential to verify the algorithm's overall performance.

4 Experiments and results

4.1 Parameter settings

To ensure reproducibility, the detailed parameter settings are provided as follows: The simulations were performed on a Windows 10 PC equipped with an i5-10400F 2.90 GHz central processing unit. The maximum number of generations, G, was set to 300. To ensure sufficient global exploration in the constrained optimization, the initial population size $\mathrm{NP}^{\mathrm{init}}$ was set to 900. The proportion p for selecting $\boldsymbol{x}_{\mathrm{pbest},g}$ was set to 5%. To efficiently eliminate low-quality solutions, the reduction rate λ and the last population percentage PPC_G of the SDE are set to 0.5 and 2%, respectively. The **ARP** size is set to 20 to ensure efficient synergistic adaptation under population reduction.

The algorithms used in this paper include the baseline

algorithm DE (Storn and Price, 1997), and the state-of-the-art (SOTA) algorithms: success-history based adaptive differential evolution algorithm including linear population size reduction (L-SHADE) (Chen et al., 2021), improved compression factor particle swarm optimization algorithm (ICFPSO) (Li and Yu, 2023), improved whale optimization algorithm (IWOA) (Bao et al., 2024) and adaptive differential evolution with optional external archive and multiple mutation strategies (MMS-JADE) (Hu et al., 2024). The remaining settings are detailed in Table 2.

Table 2 Algorithm parameters setting

Algorithm	Parameters	Values
DE	Crossover factor CR	0.3
	Mutation factor F	0.5
L-SHADE	Last population size NP_G	18
	Size of history experience set	20
ICFPSO	Inertia weights ω_{min}, ω_{max}	0.4, 0.9
	Standard deviation σ	0.3
	Learning factors c_1, c_2	2
IWOA	Spiral shape constant b	1
MMS-JADE	Interval constant N	50
	Constant factor c	0.1
SDE	Size of **ARP**	20
	Reduction rate λ	0.5
	Last population size PPC_G (%)	2%

4.2 Benchmark function experiments

The effectiveness of the algorithms employed in this study is assessed based on nine benchmark functions. As shown in Table 3, these functions are categorized into three groups: 1) Unimodal functions (f_1–f_3), which primarily assess the local exploitation capabilities; 2) Multimodal functions (f_4–f_6), designed to evaluate global exploration abilities and the capacity to escape local optima; and 3) Multimodal functions with fixed dimensions (f_7–f_9), which provide controlled evaluations of robustness and adaptability across different problem scales. To reduce randomness, each algorithm was independently executed 100 times for each benchmark.

The results are presented in Table 4 and Table 5, where the +/−/= notation is used to compare the performance of SDE against compared methods, with + indicating superior performance, −representing inferior results, and = denoting comparable outcomes. For f_1–f_3, SDE consistently achieves optimal solutions with lower standard deviations, due to the evolutionary guidance that improves the convergence quality by refining the search direction. For f_4–f_6, SDE reliably locates the global optima with stable performances, which is attributed to the synergistic evolution

Table 3 Benchmark functions

Functions	D	Limit
$f_1(x)=\sum_{i=1}^{n-1}\left[100\left(x_{i+1}-x_i^2\right)^2+\left(x_i-1\right)^2\right]$	30	$[-30, 30]^D$
$f_2(x)=\sum_{i=1}^{n}\left(\left[x_i+0.5\right]\right)^2$	30	$[-100, 100]^D$
$f_3(x)=\sum_{i=1}^{n} ix_i^4+\text{random}[0, 1)$	30	$[-1.28, 1.28]^D$
$f_4(x)=\sum_{i=1}^{n}-x_i\sin\left(\sqrt{\lvert x_i\rvert}\right)$	30	$[-500, 500]^D$
$f_5(x)=\sum_{i=1}^{n}\left[x_i^2-10\cos\left(2\pi x_i\right)+10\right]$	30	$[-5.12, 5.12]^D$
$f_6(x)=\frac{\pi}{n}10\sin^2\left(\pi y_1\right)+\frac{\pi}{n}\sum_{i=1}^{n-1}\left(y_i-1\right)\left[1+10\sin^2\left(\pi y_{i+1}\right)\right]+\frac{\pi}{n}\left(y_n-1\right)^2+\sum_{i=1}^{n}u\left(x_i, 10, 100, 4\right)$, $y_i=1+\frac{x_i+1}{4}$, $u\left(x_i, a, k, m\right)=\begin{cases}k\left(x_i-a\right)^m & x_i>a\\ 0 & -a<x_i<a\\ k\left(-x_i-a\right)^m & x_i<-a\end{cases}$	30	$[-50, 50]^D$
$f_7(x)=\sum_{i=1}^{11}\left[a_i-\frac{x_1\left(b_i^2+b_ix_2\right)}{b_i^2+b_ix_3+x_4}\right]^2$	4	$[-5, 5]^D$
$f_8(x)=-\sum_{i=1}^{4}c_i\exp\left(-\sum_{j=1}^{6}a_{ij}\left(x_j-p_{ij}\right)^2\right)$	6	$[0, 1]^D$
$f_9(x)=-\sum_{i=1}^{5}\left[\left(x-a_i\right)\left(x-a_i\right)^T+c_i\right]^{-1}$	4	$[0, 10]^D$

mechanism that enhances the global exploration capabilities. For f_7–f_9, SDE demonstrates superior or at least competitive performance in attaining the optimal solutions, highlighting its adaptability across diverse tasks. Furthermore, as shown in the last row of Table 4, the Wilcoxon rank-sum results show that all p-values are less than 0.05, indicating statistically significant differences between SDE and the other algorithms.

4.3 Path planning experiments

Three path planning experiments are performed to validate the competitive advantages of SDE over the SOTA algorithms in this section. The simulation space was set to 6 000 m × 6 000 m × 450 m. The number of waypoints was n = 12, and the number of cubic spline insertion points was m = 8. The weight of the fitness function was $\boldsymbol{\omega}$ = [1, 8, 5], and the AUV velocity relative to the flow field in the forward direction was u_{const} = 1.5 m/s. The drag coefficient was α = [3, 15], and the terrain redundancy safety distance was d_{safe} = 20 m. The remaining simulation configurations

Table 4 Average values of the benchmark functions

Function	DE	ICFPSO	IWOA	MMS-JADE	L-SHADE	SDE
f_1	4.8982×10^{2}	7.9971×10^{3}	2.5610×10^{1}	2.2406×10^{1}	1.9889×10^{1}	1.1099×10^{-1}
f_2	8.3135×10^{-1}	9.1934×10^{1}	1.0638×10^{-2}	1.1682×10^{-11}	1.8275×10^{-13}	2.8503×10^{-15}
f_3	5.6778×10^{-2}	1.2232×10^{-1}	3.4754×10^{-4}	3.3104×10^{-3}	3.3943×10^{-3}	3.3468×10^{-4}
f_4	-8.5397×10^{3}	-8.4278×10^{3}	-9.1123×10^{3}	-6.7993×10^{3}	-9.8442×10^{3}	-1.2452×10^{4}
f_5	1.1283×10^{2}	6.4683×10^{1}	1.6118×10^{1}	4.0365×10^{1}	6.9147×10^{1}	9.5239×10^{-1}
f_6	4.3474×10^{-1}	8.3332×10^{0}	3.3825×10^{-3}	6.9112×10^{-3}	6.9009×10^{-14}	3.9108×10^{-16}
f_7	4.9642×10^{-4}	3.9753×10^{-4}	4.3349×10^{-4}	3.7769×10^{-4}	3.0749×10^{-4}	3.0749×10^{-4}
f_8	-3.3220×10^{0}	-3.2447×10^{0}	-3.2425×10^{0}	-3.2622×10^{0}	-3.3220×10^{0}	-3.3220×10^{0}
f_9	-1.0153×10^{1}	-8.2697×10^{0}	-9.3545×10^{0}	-1.0153×10^{1}	-1.0153×10^{1}	-1.0153×10^{1}
+\−\=	9\0\0	9\0\0	9\0\0	8\0\1	7\0\2	
p	1.0550×10^{-99}	1.0550×10^{-99}	1.5624×10^{-70}	7.7193×10^{-86}	9.1112×10^{-99}	

Table 5 Standard deviation values of the benchmark functions

Function	DE	ICFPSO	IWOA	MMS-JADE	L-SHADE	SDE
f_1	6.6712×10^{1}	8.7483×10^{3}	5.6755×10^{-1}	9.4006×10^{0}	5.9309×10^{-1}	3.9943×10^{-3}
f_2	9.5809×10^{-2}	7.5181×10^{1}	5.4460×10^{-2}	5.2137×10^{-12}	4.1561×10^{-13}	2.2315×10^{-16}
f_3	1.0465×10^{-2}	5.0231×10^{-2}	3.3716×10^{-4}	9.3425×10^{-4}	1.0071×10^{-3}	9.6865×10^{-5}
f_4	2.4045×10^{2}	7.5111×10^{2}	1.7143×10^{3}	7.0067×10^{2}	2.9522×10^{2}	1.0940×10^{2}
f_5	7.9928×10^{0}	2.3053×10^{1}	4.0898×10^{1}	1.6505×10^{1}	7.5509×10^{0}	2.9158×10^{-1}
f_6	7.7950×10^{-2}	5.3059×10^{0}	5.7626×10^{-3}	2.7209×10^{-2}	5.0218×10^{-13}	2.9876×10^{-16}
f_7	5.3251×10^{-5}	2.6035×10^{-4}	3.1454×10^{-4}	2.4363×10^{-4}	4.3148×10^{-19}	2.7105×10^{-19}
f_8	1.2868×10^{-11}	5.6708×10^{-2}	5.6212×10^{-2}	5.9445×10^{-2}	2.6661×10^{-15}	2.6580×10^{-15}
f_9	2.5494×10^{-8}	2.4903×10^{0}	1.8531×10^{0}	8.9143×10^{-15}	9.7630×10^{-15}	8.0926×10^{-15}
+\−\=	9\0\0	9\0\0	9\0\0	9\0\0	9\0\0	

are detailed in Table 6. To ensure the objectivity of the experimental results, each algorithm was independently executed 100 times.

4.3.1 Case 1

The environment in Case 1 is characterized by a regular and smooth flow field, as shown in Figure 5. However, the presence of multiple restricted regions and uneven terrain significantly reduces the available navigable space, resulting in multiple similar local optima. Therefore, Case 1 primarily evaluates the global exploration capability.

The simulation results are presented in Figures 5 to 7 to give a representative assessment of each algorithm's performance. In Case 1, the path planned by SDE offers several advantages. Compared to MMS-JADE, SDE prevents redundant horizontal movements and produces a shorter path. Relative to DE, ICFPSO, and IWOA, the SDE path is smoother and more stable, minimizing unnecessary heave motion. In contrast to L-SHADE, SDE effectively avoids local optima, fully leverages the flow field, and selects the most energy-efficient solution.

The box plots in Figure 8 illustrate the distribution of fitness values and their medians across 100 runs for each algorithm in Case 1. SDE exhibits superior global optimization capability by achieving the lowest median fitness value. Simultaneously, SDE exhibits relatively low worst-case and outlier values, highlighting its ability to avoid poor solutions.

Figure 9 presents the convergence curves based on the median results, while Figure 10 illustrates variations of the average fitness across generations during optimization. SDE demonstrates a rapid decline in the early generations, reflecting a strong global search capability. The final convergence value of SDE is lower than those of the SOTA algorithms, further validating its superior global optimization capability.

4.3.2 Case 2

As shown in Figure 11, Case 2 presents a larger free space between the terminal points. However, the more complex flow field introduces greater environmental variability. This case primarily evaluates the algorithms' stability and convergence performance. As shown in Figure 12, SDE effectively leverages the flow field to reduce energy consumption and minimize travel in cross-current regions, resulting in a more efficient path. Furthermore, as shown in Figure 13, SDE avoids unnecessary heave motion, contributing to improved motion stability and smoother navi-

Table 6 Simulation configurations

Cases	Parameters	Values
C1	$\boldsymbol{P}_S, \boldsymbol{P}_G$	{(−2 700, 1 200, 300), (2 400, −2 100, 300)}
	$\boldsymbol{x}^v$	[1 526, 1 761, −1 252, 46, −1 572, −1 844, 1 852, −2 600]
	$\boldsymbol{y}^v$	[−1 825, 2 849, 451, 499, 2 700, −2 400, 2 272, 2 349]
	$\boldsymbol{\Gamma}$	[−1 430, −1 665, 731, −1 598, −1 917, −673, −488, −2 051]
	$\boldsymbol{\delta}$	[592, 523, 743, 356, 709, 583, 622, 439]
	$\boldsymbol{P}_R$	{(−1 500, 100), (800, −1 100), (−1 200, −1 800), (0, 800)}
	$\boldsymbol{r}_R$	[600, 600, 600, 600]
	$\boldsymbol{x}^h$	[1 260, 115, 879, 231, 1 489, −463, −1 672, −2 686, 2 647, −1 201]
	$\boldsymbol{y}^h$	[2 182, −2524, −1 059, 595, 127, 1 730, −2 323, −891, −837, −945]
	$\boldsymbol{x}^a = \boldsymbol{y}^a$	[1 032, 1 636, 1 247, 1 510, 526, 690, 667, 1 919, 1 305, 948]
	$\boldsymbol{h}$	[250, 176, 237, 175, 100, 147, 184, 158, 114, 139]
C2	$\boldsymbol{P}_S, \boldsymbol{P}_G$	{(−1 600, 2 700, 250), (800, −2 700, 200)}
	$\boldsymbol{x}^v$	[262, 2 317, −1 444, −676, 1 404, 2 611, 99, 2 923]
	$\boldsymbol{y}^v$	[2 144, 2 002, 805, −821, 3 958, 1 885, −1 693, −991]
	$\boldsymbol{\Gamma}$	[85, 1 805, −2 276, 852, −1 527, −789, 1 128, −1 969]
	$\boldsymbol{\delta}$	[597, 353, 651, 580, 312, 729, 548, 376]
	$\boldsymbol{P}_R$	{(−1 200, 900), (300, −500), (−1 500, −1 500), (1 200, 1 200), (2 100, −1 500)}
	$\boldsymbol{r}_R$	[650, 500, 600, 500, 600]
	$\boldsymbol{x}^h$	[560, −493, −2 678, 1 564, −759, −155, 855, 1 791, 583, −1 224]
	$\boldsymbol{y}^h$	[2 469, 2 273, 1 210, 616, −1 137, −989, −398, −1 609, −160, 1 993]
	$\boldsymbol{x}^a = \boldsymbol{y}^a$	[997, 1 689, 1 632, 1 124, 737, 1 967, 1 876, 1 363, 500, 670]
	$\boldsymbol{h}$	[51, 52, 50, 130, 84, 198, 139, 106, 54, 64]
C3	$\boldsymbol{P}_S, \boldsymbol{P}_G$	{(−2 700, −2 700, 100), (2 100, 2 700, 300)}
	$\boldsymbol{x}^v$	[−1 200, −2 000, −800, 1 400, 1 200, 1 200, −2 800, −800, 2 400, −1 000]
	$\boldsymbol{y}^v$	[1 600, −1 600, 600, −200, 2 000, 1 000, −800, −800, −1 400, 2 600]
	$\boldsymbol{\Gamma}$	[−1 854, 2 388, −2 038, 2 198, −2 274, 1 734, −2 322, 1 684, −2 196, 2 184]
	$\boldsymbol{\delta}$	[580, 490, 438, 516, 646, 694, 658, 426, 502, 470]
	$\boldsymbol{P}_R$	[(−1 800, −1 800), (400, 1 600), (−500, −300), (−1 800, 1 000), (1 500, 200)]
	$\boldsymbol{r}_R$	[800, 800, 550, 700, 500]
	$\boldsymbol{x}^h$	[1 709, 650, 732, −2 371, −35, −1 767, 726, 2 384]
	$\boldsymbol{y}^h$	[−142, 2 422, −2 269, 2 635, 1 070, −2 085, 1 898, 2 192]
	$\boldsymbol{x}^a = \boldsymbol{y}^a$	[777, 746, 1 733, 1 189, 1 585, 1 000, 1 517, 1 245]
	$\boldsymbol{h}$	[108, 185, 170, 146, 297, 169, 59, 71]

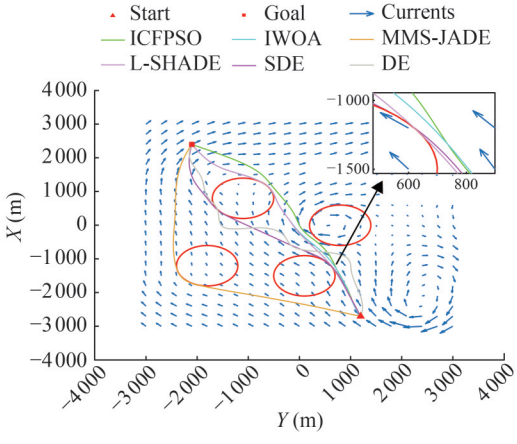


Figure 5 Top view of the median optimal solution for Case 1. The inset shows an expanded view of the area marked by the black arrow

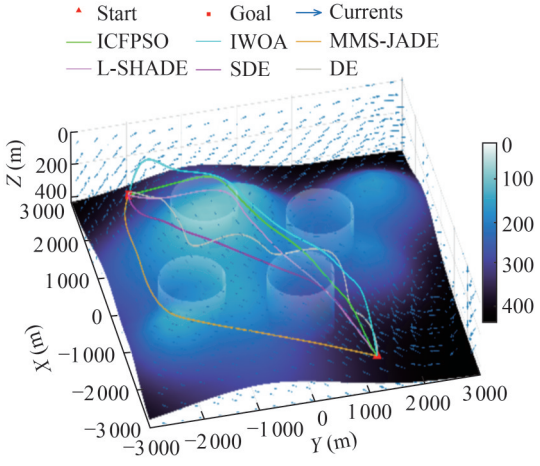


Figure 6 Side view of the median optimal solution for Case 1

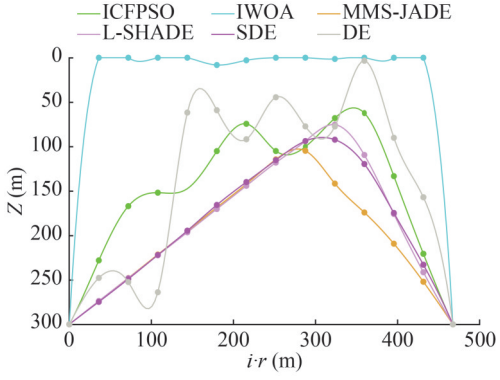


Figure 7 Vertical projection of the median optimal path in Case 1

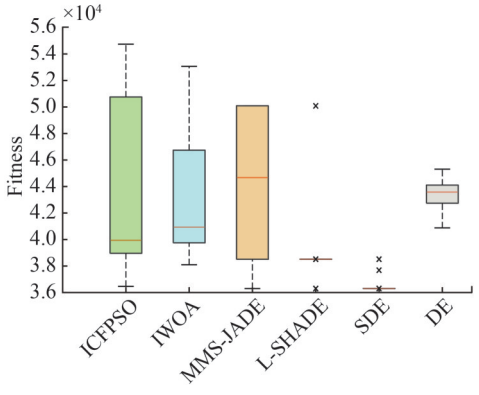


Figure 8 Box plots of fitness results for Case 1. Red lines and the "×" symbols indicate the medians and outliers, respectively

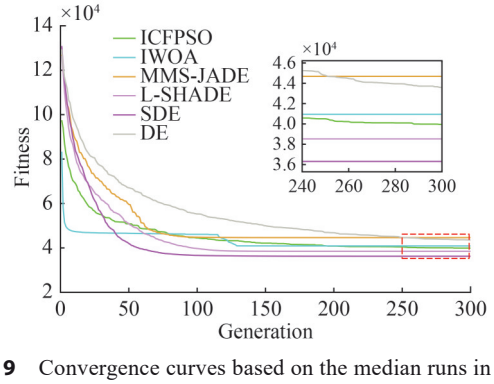


Figure 9 Convergence curves based on the median runs in Case 1. The inset shows an expanded view of the area marked by the dashed line

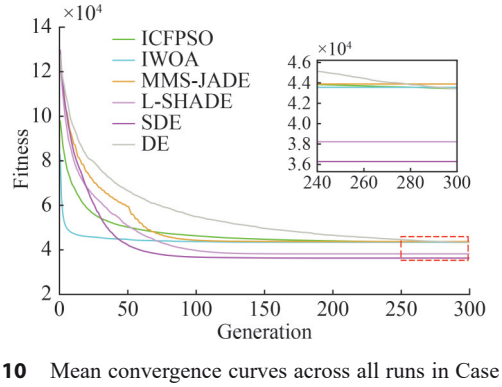


Figure 10 Mean convergence curves across all runs in Case 1. The inset shows an expanded view of the area marked by the dashed line

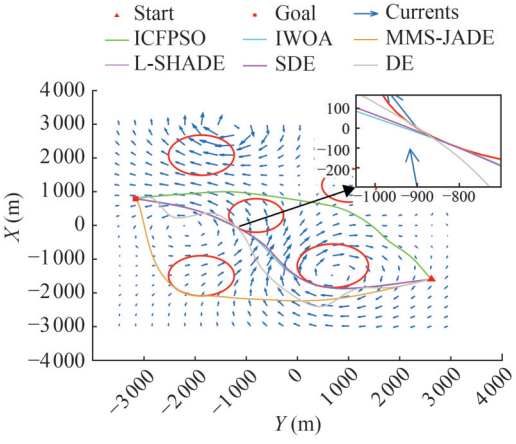


Figure 11 Top view of the median optimal solution for Case 2. The inset shows an expanded view of the area marked by the black arrow

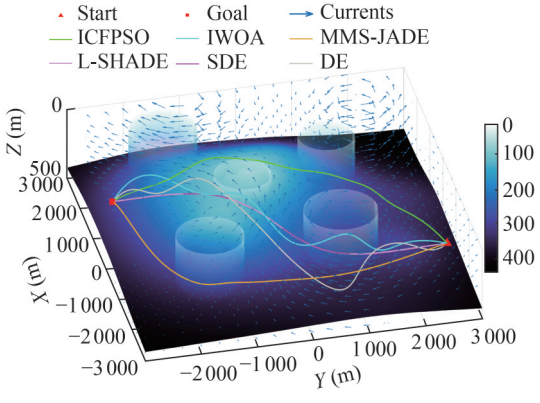


Figure 12 Side view of the median optimal solution for Case 2

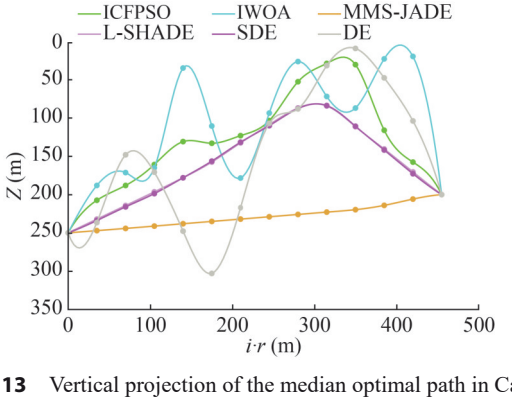


Figure 13 Vertical projection of the median optimal path in Case 2

gation. Notably, L-SHADE produced a path like that of SDE, suggesting a consistent global optimal solution for Case 2 across different algorithms.

As shown in Figure 14, SDE exhibits superior stability and convergence by producing fewer outliers, all of which remain at relatively small fitness values. Simultaneously, although the box plot indicates that both SDE and L-SHADE achieved the lowest median fitness, the SDE results were more concentrated, reflecting better consistency and convergence.

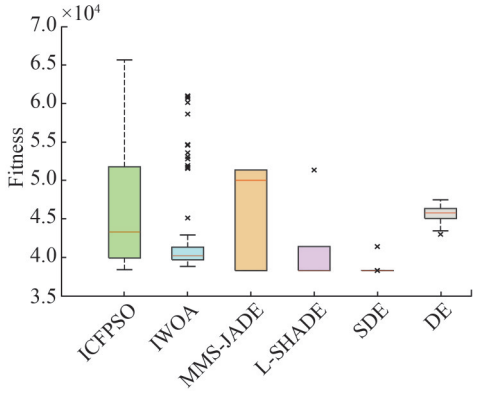


Figure 14 Box plots of fitness results for Case 2. Red lines and the "×" symbols indicate the medians and outliers, respectively

The SDE convergence curves in Figures 15 and 16 show similar trends, demonstrating consistent convergence behavior across multiple runs. This consistency indicates the robust performance and reliable optimization capability of SDE. Moreover, the SDE convergence process is smooth and free from significant variation, implying stable optimization. The SDE fitness converges to the global minimum first after approximately 80 generations, reflecting its superior convergence efficiency.

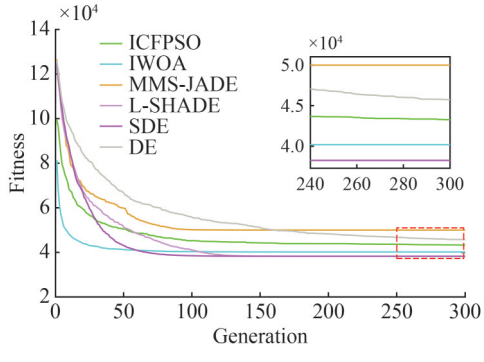


Figure 15 Convergence curves based on the median runs in Case 2. The inset shows an expanded view of the area marked by the dashed line

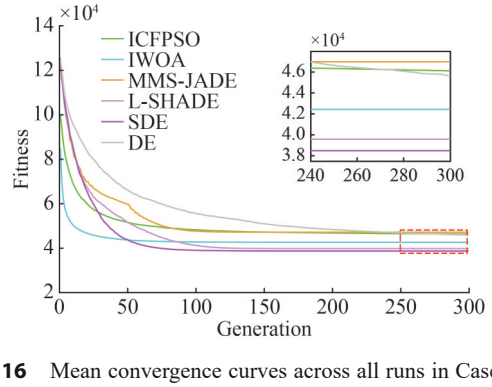


Figure 16 Mean convergence curves across all runs in Case 2. The inset shows an expanded view of the area marked by the dashed line

4.3.3 Case 3

Case 3 presents the most challenging environment among all scenarios, as shown in Figure 17. The coexistence of restricted regions, irregular terrain, and complex flow conditions poses significant challenges to both optimization and solution stability (Sun et al., 2022). Therefore, this scenario offers a comprehensive evaluation of the overall algorithm performance.

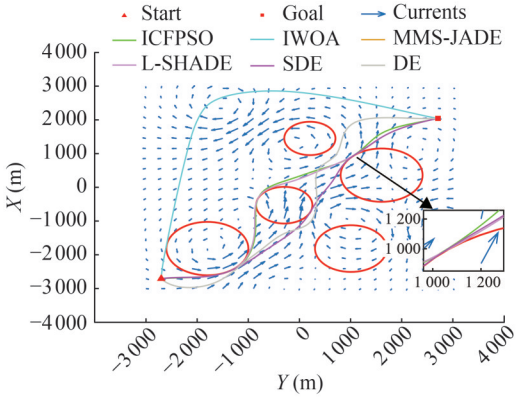


Figure 17 Top view of the median optimal solution for Case 3. The inset shows an expanded view of the area marked by the black arrow

As illustrated in Figures 18 and 19, SDE successfully navigates the narrow passage in the complex flow field, resulting in the most energy-efficient trajectory. These results highlight the superior performance of SDE in dealing with multiple constraints and energy consumption simultaneously. The SDE algorithm balances obstacle avoidance, terrain safety, and flow-field use, demonstrating strong global search capability and robust adaptability in complex environments.

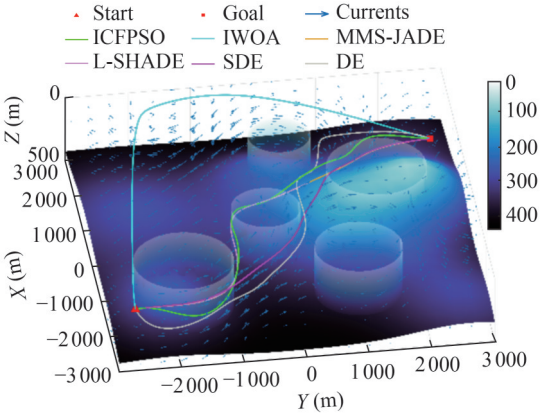


Figure 18 Side view of the median optimal solution for Case 3

As shown in Figure 20, the overall performance of SDE stands out in several respects. It achieves the lowest median fitness among all compared methods, with fitness values concentrated near the global optimum, reflecting its strong algorithmic stability and global optimization ability.

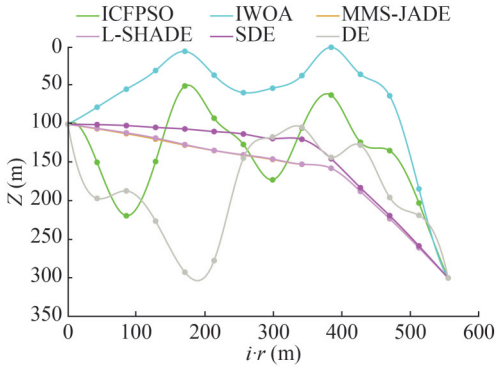


Figure 19 Vertical projection of the median optimal path in Case 3

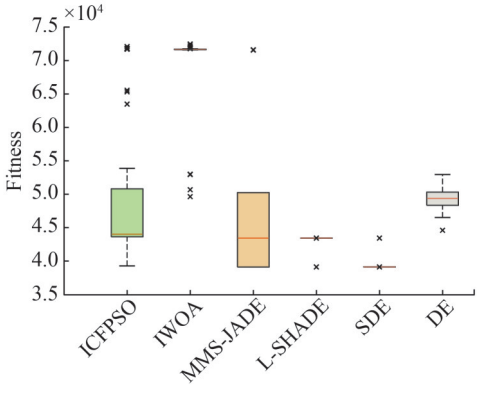


Figure 20 Box plots of fitness results in Case 3. Red lines and the "×" symbols indicate the medians and outliers, respectively

Furthermore, the limited extent of outlier values suggests that SDE maintains robustness against performance degradation under complex conditions.

The convergence curves shown in Figures 21 and 22 demonstrate the excellent overall performance of SDE. In the early stages, SDE exhibits a rapid decline in fitness within the first 50 generations, indicating a strong global search capability. The convergence process remains smooth and free from significant fluctuations, and SDE converges first to the global minimum at approximately the 80th generation. Its final convergence value surpasses those of the SOTA algorithms, validating its effectiveness in reaching high-quality solutions. Moreover, the convergence trends for both median and average results are highly consistent, demonstrating robust and stable performance over multiple runs. These results confirm that SDE possesses superior global optimization ability, efficient convergence behavior, and strong stability.

4.3.4 Metric analyses

The metrics include best fitness m_{best}, average fitness m_{mean}, median fitness m_{median}, standard deviation m_{std}, convergence quality m_{convg}, average runtime per run m_t, average constraint penalty m_c, total score m_{score}, and final rank m_{rank}. The m_c metric records the average number of generations at which the fitness value first falls below 1.05 times

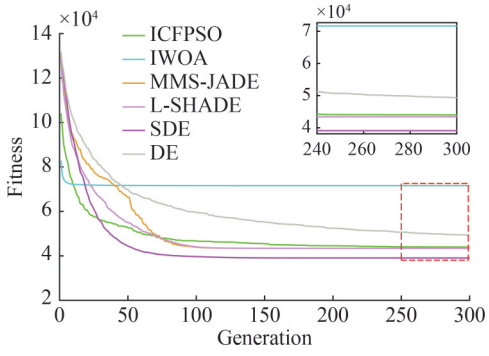


Figure 21 Convergence curves based on median run in Case 3. The inset shows an expanded view of the area marked by the dashed line

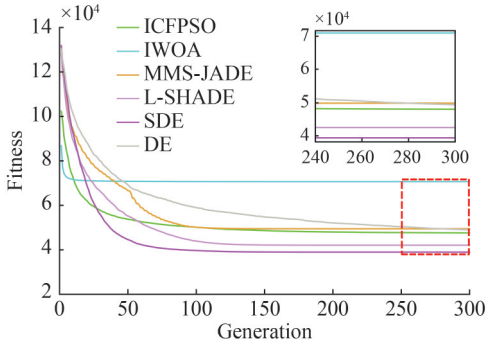


Figure 22 Mean convergence curves across all runs in Case 3. The inset shows an expanded view of the area marked by the dashed line

the global optimum. For the m_{score}, each metric awards 5 points to the algorithm ranked first, 4 points for second place, and so on. In the event of a tie, the same score is assigned. The detailed results are summarized in Table 7, with the best-performing values highlighted in bold.

The values in Table 7 indicate that, compared with the four other SOTA algorithms, the m_{best}, m_{mean} and m_{median} of SDE improve by averages of 2.37%, 15.99% and 13.42% across all cases, respectively, indicating better optimization and reduced risk of local optima. Furthermore, the m_{std} and m_{convg} of SDE are significantly reduced by averages of 80.69% and 66.65%, respectively, indicating enhanced stability and convergence efficiency. In addition, the SDE computational m_t is reduced by averages of 39.91%, indicating improved computational efficiency. Meanwhile, the m_{std}-to-m_{best} ratio of SDE for the three cases is 1.50%, 1.95%, and 2.17%, respectively, indicating its high stability and adaptability in increasingly complex environments. Notably, the m_c of SDE remains below one in all cases, highlighting its ability to ensure compliance with various constraints while optimizing energy consumption. Overall, SDE achieves superior optimization with high stability, efficiency, and strong constraint-handling capabilities.

4.4 Ablation experiments

The ablation experiments evaluate the contribution of

Table 7 Simulation results

Case	Metrics	ICFPSO	IWOA	MMS-JADE	L-SHADE	SDE
C1	m_{best}	3.6459×10^{4}	3.8092×10^{4}	$\mathbf{3.6295 \times 10^{4}}$	$\mathbf{3.6295 \times 10^{4}}$	$\mathbf{3.6295 \times 10^{4}}$
	m_{mean}	4.3421×10^{4}	4.3554×10^{4}	4.3873×10^{4}	3.8338×10^{4}	$\mathbf{3.6442 \times 10^{4}}$
	m_{median}	3.9928×10^{4}	4.0910×10^{4}	4.4667×10^{4}	3.8510×10^{4}	$\mathbf{3.6295 \times 10^{4}}$
	m_{std}	5.9125×10^{3}	4.0977×10^{3}	5.2579×10^{3}	1.4024×10^{3}	$\mathbf{5.4314 \times 10^{2}}$
	m_{convg}	2.8008×10^{2}	2.9756×10^{2}	2.6346×10^{2}	2.7497×10^{2}	$\mathbf{8.5260 \times 10^{1}}$
	m_t (s)	1.2764×10^{1}	1.2794×10^{1}	1.3496×10^{1}	7.3971×10^{0}	$\mathbf{6.3483 \times 10^{0}}$
	m_c	7.5874×10^{1}	9.4726×10^{2}	3.7477×10^{-3}	$\mathbf{2.1621 \times 10^{-5}}$	3.1431×10^{-3}
	m_{score}	16	12	17	29	**34**
	m_{rank}	4	5	3	2	**1**
C2	m_{best}	3.8393×10^{4}	3.8823×10^{4}	$\mathbf{3.8281 \times 10^{4}}$	$\mathbf{3.8281 \times 10^{4}}$	$\mathbf{3.8281 \times 10^{4}}$
	m_{mean}	4.6092×10^{4}	4.2422×10^{4}	4.6967×10^{4}	3.9567×10^{4}	$\mathbf{3.8470 \times 10^{4}}$
	m_{median}	4.3277×10^{4}	4.0177×10^{4}	5.0025×10^{4}	$\mathbf{3.8281 \times 10^{4}}$	$\mathbf{3.8281 \times 10^{4}}$
	m_{std}	6.9114×10^{3}	5.5602×10^{3}	5.5767×10^{3}	1.9234×10^{3}	$\mathbf{7.4496 \times 10^{2}}$
	m_{convg}	2.5713×10^{2}	1.8102×10^{2}	2.4469×10^{2}	1.7175×10^{2}	$\mathbf{8.1190 \times 10^{1}}$
	m_t (s)	1.2747×10^{1}	1.2690×10^{1}	1.2737×10^{1}	7.0787×10^{0}	$\mathbf{6.7475 \times 10^{0}}$
	m_c	1.2347×10^{1}	1.0933×10^{3}	3.1754×10^{-3}	$\mathbf{3.4654 \times 10^{-7}}$	2.6813×10^{-4}
	m_{score}	11	17	16	31	**34**
	m_{rank}	5	3	4	2	**1**
C3	m_{best}	3.9284×10^{4}	4.9650×10^{4}	$\mathbf{3.9122 \times 10^{4}}$	$\mathbf{3.9122 \times 10^{4}}$	$\mathbf{3.9122 \times 10^{4}}$
	m_{mean}	4.7911×10^{4}	7.0908×10^{4}	4.9755×10^{4}	4.2403×10^{4}	$\mathbf{3.9295 \times 10^{4}}$
	m_{median}	4.4022×10^{4}	7.1676×10^{4}	4.3440×10^{4}	4.3440×10^{4}	$\mathbf{3.9122 \times 10^{4}}$
	m_{std}	8.5298×10^{3}	3.9802×10^{3}	1.2718×10^{4}	1.8532×10^{3}	$\mathbf{8.5037 \times 10^{2}}$
	m_{convg}	2.7082×10^{2}	3.0000×10^{2}	2.4358×10^{2}	2.5477×10^{2}	$\mathbf{8.0270 \times 10^{1}}$
	m_t (s)	1.4083×10^{1}	1.1840×10^{1}	1.3533×10^{1}	7.2208×10^{0}	$\mathbf{6.3730 \times 10^{0}}$
	m_c	3.4275×10^{1}	4.9978×10^{1}	4.1073×10^{0}	8.1936×10^{0}	$\mathbf{4.4604 \times 10^{-1}}$
	m_{score}	14	11	22	27	**35**
	m_{rank}	4	5	3	2	**1**

each component in the proposed algorithm. Specifically, DE serves as a baseline. SDE-1 is a single population variant that excludes $\boldsymbol{P}_{aid}$. SDE-2 removes the evolutionary guidance from the mutation operator. SDE-3 omits the population reduction strategy.

The results of the ablation experiments are presented in Figure 23 and Table 8. First, SDE-1 yields suboptimal median values for Case 1 and Case 3, as shown in Figure 23. Notably, its m_{mean} and m_{median} increase significantly by averages of 5.16% and 5.71% across all cases compared to SDE. This shows that the adaptive synergistic evolution mechanism plays a crucial role in improving the optimization and reducing the risk of falling into local optima.

Second, as shown in Figure 23, the fitness value distribution of SDE-2 is significantly more dispersed than that of SDE. The m_{std} and m_{convg} of SDE-2 increase significantly by averages of 96.29% and 80.58% compared to SDE, respectively. This indicates that the weighted evolutionary guidance plays a crucial role in improving the algorithmic stability and convergence quality.

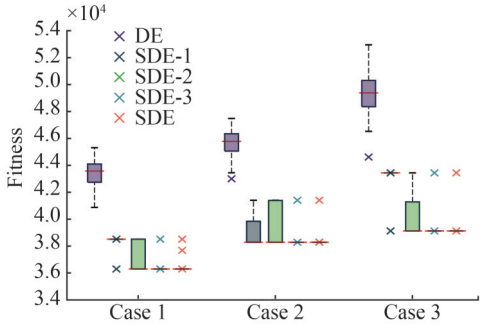


Figure 23 Ablation experiments. In each case, from left to right, the results correspond to DE, SDE-1, SDE-2, SDE-3, and SDE. Red lines and the "×" symbols indicate the medians and outliers, respectively

Finally, as shown in Figure 23 and Table 8, the overall optimization performance of SDE-3 is comparable to that of SDE. Notably, SDE reduces the m_t by an average of 62.91% compared to SDE-3. This advantage is attributed

Table 8 Ablation experiment results

Case	Metrics	DE	SDE-1	SDE-2	SDE-3	SDE
C1	m_{best}	4.0873×10^{4}	3.6295×10^{4}	3.6295×10^{4}	3.6295×10^{4}	3.6295×10^{4}
	m_{mean}	4.3408×10^{4}	3.8200×10^{4}	3.6996×10^{4}	3.6450×10^{4}	3.6442×10^{4}
	m_{median}	4.3567×10^{4}	3.8510×10^{4}	3.6296×10^{4}	3.6295×10^{4}	3.6295×10^{4}
	m_{std}	9.8810×10^{2}	7.7261×10^{2}	9.8343×10^{2}	5.6812×10^{2}	5.4314×10^{2}
	m_{convg}	3.0000×10^{2}	2.7325×10^{2}	1.6167×10^{2}	8.7290×10^{1}	8.5260×10^{1}
	m_t (s)	1.2480×10^{1}	6.7990×10^{0}	1.0355×10^{1}	1.7102×10^{1}	6.3483×10^{0}
	m_c	1.5952×10^{3}	2.5022×10^{-5}	9.2251×10^{-3}	4.1257×10^{-5}	3.1431×10^{-3}
C2	m_{best}	4.2999×10^{4}	3.8281×10^{4}	3.8281×10^{4}	3.8281×10^{4}	3.8281×10^{4}
	m_{mean}	4.5661×10^{4}	3.9062×10^{4}	3.9124×10^{4}	3.8500×10^{4}	3.8470×10^{4}
	m_{median}	4.5765×10^{4}	3.8281×10^{4}	3.8281×10^{4}	3.8281×10^{4}	3.8281×10^{4}
	m_{std}	9.1206×10^{2}	1.3587×10^{3}	1.3919×10^{3}	8.0058×10^{2}	7.4496×10^{2}
	m_{convg}	3.0000×10^{2}	1.4593×10^{2}	1.5150×10^{2}	8.1150×10^{1}	8.1190×10^{1}
	m_t (s)	1.3074×10^{1}	7.0013×10^{0}	1.0366×10^{1}	1.8677×10^{1}	6.7475×10^{0}
	m_c	1.6190×10^{3}	4.0222×10^{-7}	3.6258×10^{-4}	2.0390×10^{-5}	2.6813×10^{-4}
C3	m_{best}	4.4613×10^{4}	3.9122×10^{4}	3.9122×10^{4}	3.9122×10^{4}	3.9122×10^{4}
	m_{mean}	4.9270×10^{4}	4.2878×10^{4}	4.0202×10^{4}	3.9381×10^{4}	3.9295×10^{4}
	m_{median}	4.9359×10^{4}	4.3440×10^{4}	3.9122×10^{4}	3.9122×10^{4}	3.9122×10^{4}
	m_{std}	1.4528×10^{3}	1.4592×10^{3}	1.8789×10^{3}	1.0305×10^{3}	8.5037×10^{2}
	m_{convg}	3.0000×10^{2}	2.7563×10^{2}	1.3287×10^{2}	8.8270×10^{1}	8.0270×10^{1}
	m_t (s)	1.2836×10^{1}	6.8420×10^{0}	9.7896×10^{0}	1.6761×10^{1}	6.3730×10^{0}
	m_c	1.6220×10^{3}	9.3808×10^{0}	2.6969×10^{0}	6.5122×10^{-1}	4.4604×10^{-1}

to the cosine-based population-size adjustment strategy, which reduces the computational burden of synergistic evolution while preserving the optimization performance.

Compared to the baseline DE algorithm, SDE exhibits significant improvements across all metrics. Specifically, the m_{best}, m_{mean}, m_{median}, m_{std}, m_{convg}, and m_t are improved by averages of 11.49%, 17.35%, 17.93%, 34.94%, 72.59%, and 49.29%, respectively. Moreover, the average constraint penalty (m_c) of DE reaches around 1 600, whereas SDE maintains this value below one in all cases. This result confirms that SDE not only enhances optimization quality and computational efficiency but also demonstrates strong constraint-handling ability, ensuring safe and reliable solutions across diverse environments.

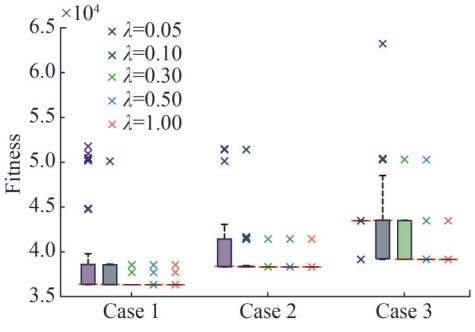


Figure 24 Box plot results for different λ configurations. In each case, from left to right, the results correspond to λ=0.05, 0.10, 0.30, 0.50, and 1.00. Red lines and "×" symbols indicate the medians and outliers, respectively

4.5 Sensitivity experiments

The sensitivity experiments evaluate the impact of the hyperparameter on algorithmic performance, focusing on different λ configurations for the cosine-based population-size adjustment strategy. The results are presented in Figure 24 and Table 9.

A larger λ slows the population-size reduction, preserving more individuals, which enhances population diversity and global search performance. However, this increases the computational burden. Compared to λ = 0.05, when λ = 1.00, the computational m_t increases significantly by an average of 577.02%. In contrast, a smaller λ accelerates the population reduction, promoting local exploitation by pruning the low-quality solutions. However, if λ is too small, the population diversity declines, leading to reduced stability and a higher risk of premature convergence. Compared to λ = 0.50, when λ = 0.05, the m_{mean}, m_{median}, m_{std}, and m_{convg} increase significantly by averages of 5.43%, 3.81%, 312.13%, and 115.23% across all cases, respectively. Therefore, λ should be kept above a certain threshold, with a recommended minimum value of 0.50.

Table 9 Results for different λ configurations

Case	Metric	λ=0.05	λ=0.10	λ=0.30	λ=0.50	λ=1.00
C1	m_{best}	3.6304×10^{4}	3.6298×10^{4}	3.6295×10^{4}	3.6295×10^{4}	3.6295×10^{4}
	m_{mean}	3.8447×10^{4}	3.7405×10^{4}	3.6613×10^{4}	3.6442×10^{4}	3.6411×10^{4}
	m_{median}	3.6362×10^{4}	3.6330×10^{4}	3.6297×10^{4}	3.6295×10^{4}	3.6295×10^{4}
	m_{std}	3.9613×10^{3}	2.0973×10^{3}	7.6334×10^{2}	5.4314×10^{2}	4.7266×10^{2}
	m_{convg}	1.7762×10^{2}	1.6423×10^{2}	1.0344×10^{2}	8.5260×10^{1}	8.0490×10^{1}
	m_t (s)	1.0156×10^{0}	1.5924×10^{0}	4.1378×10^{0}	6.3483×10^{0}	9.6820×10^{0}
	m_c	3.7074×10^{1}	8.6255×10^{0}	8.4011×10^{-3}	3.1431×10^{-3}	8.6994×10^{-5}
C2	m_{best}	3.8284×10^{4}	3.8283×10^{4}	3.8282×10^{4}	3.8281×10^{4}	3.8281×10^{4}
	m_{mean}	3.9656×10^{4}	3.9273×10^{4}	3.8441×10^{4}	3.8470×10^{4}	3.8375×10^{4}
	m_{median}	3.8363×10^{4}	3.8306×10^{4}	3.8283×10^{4}	3.8281×10^{4}	3.8281×10^{4}
	m_{std}	2.7349×10^{3}	2.7114×10^{3}	6.8364×10^{2}	7.4496×10^{2}	5.3525×10^{2}
	m_{convg}	8.7290×10^{1}	7.3640×10^{1}	6.8580×10^{1}	6.7090×10^{1}	6.3700×10^{1}
	m_t (s)	1.1044×10^{0}	1.6532×10^{0}	4.4749×10^{0}	6.7475×10^{0}	1.0355×10^{1}
	m_c	5.1507×10^{-1}	1.9683×10^{-1}	4.5884×10^{-3}	2.6813×10^{-4}	6.2105×10^{-6}
C3	m_{best}	3.9122×10^{4}	3.9131×10^{4}	3.9125×10^{4}	3.9123×10^{4}	3.9122×10^{4}
	m_{mean}	4.2403×10^{4}	4.2737×10^{4}	4.1141×10^{4}	3.9366×10^{4}	3.9252×10^{4}
	m_{median}	4.3440×10^{4}	4.3466×10^{4}	3.9151×10^{4}	3.9124×10^{4}	3.9122×10^{4}
	m_{std}	1.8532×10^{3}	3.8224×10^{3}	2.7821×10^{3}	1.3241×10^{3}	7.4019×10^{2}
	m_{convg}	2.5477×10^{2}	2.1860×10^{2}	1.6762×10^{2}	8.2920×10^{1}	7.6420×10^{1}
	m_t (s)	7.2208×10^{0}	1.0128×10^{0}	1.5024×10^{0}	4.1432×10^{0}	1.0116×10^{1}
	m_c	8.1936×10^{0}	2.0911×10^{1}	3.8253×10^{0}	4.6154×10^{-1}	3.2392×10^{-1}

4.6 Field experiment

To validate the feasibility of the proposed algorithm, a surface path-planning experiment was conducted for real AUV operations. As shown in Figure 25, the AUV's forward compartment is equipped with an attitude and heading reference system in its forward compartment, while the antenna compartment houses a Global Positioning System module and a wireless communication module.

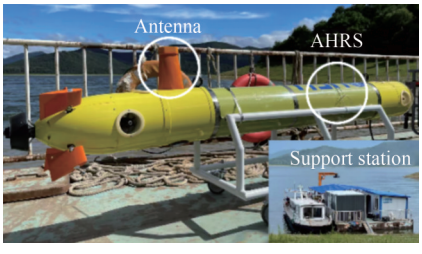


Figure 25 AUV equipped with an attitude and heading reference system used in the field experiment. The inset shows the support station setup

The experiment was conducted in a shallow waterway of Songhua Lake, Jilin, China. The map data were obtained from Tianditu at level 15 (National Geomatics Center of China, 2011), with a positional accuracy of approximately 5 m. As shown in Figure 26, the onshore

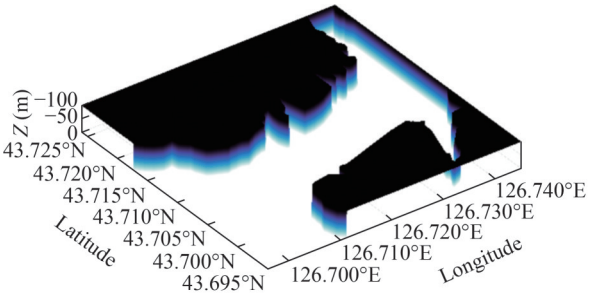


Figure 26 Onshore terrain of the field experiment location

land depth $Z_h(x, y)$ was set to −100 m in Eq. (2), while the water surface was set to 0. In addition, several restricted regions were established, as shown in Figure 27, including terrain obstacles and port operation areas. The start and end points were set to (43.722 3°N, 126.739 7°E) and (43.715 4°N, 126.697 3°E), respectively.

By inputting the initial coordinates, terrain information, and restricted regions, the SDE algorithm generated a series of target waypoints and an optimized path via an onshore support station. After verification by human operators to ensure its safety and feasibility, the planned path was then uploaded to the AUV for autonomous navigation. This deployment strategy also offloads the computational burden from the AUV.

The experimental results demonstrate that the AUV navigates smoothly to the destination while avoiding terrain obstacles and restricted regions. As shown in Figures 27 and 28, the actual trajectory and yaw angle values align with the target commands, which not only confirms the accuracy of the position and attitude reference system but also demonstrates that the proposed algorithm effectively satisfies the maneuverability constraints. These results further validate the feasibility and reliability of the proposed path-planning approach.

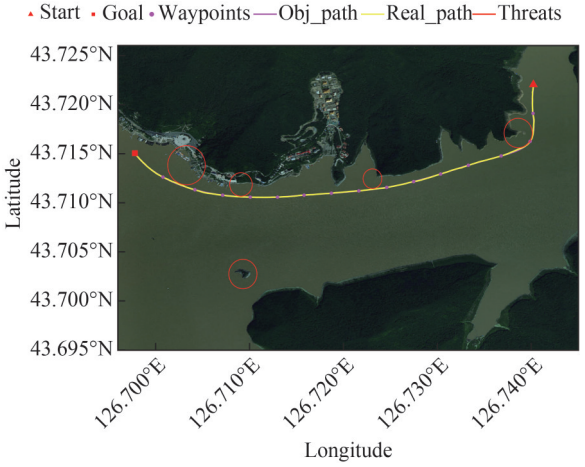


Figure 27 Field experiment results

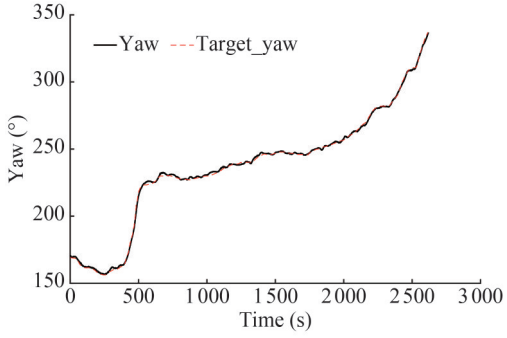


Figure 28 Yaw angle curve

5 Conclusion

In this work, an SDE-based algorithm is proposed to address the energy-efficient path-planning problem for AUV operation in complex 3D environments with multiple constraints. The proposed method incorporates three basic components: an adaptive synergistic evolution mechanism to enhance global exploration, a weighted evolutionary guidance that improves search direction, and a cosine-based strategy that adjusts the population size of the synergistic evolution.

As a result, the SDE-based path-planning algorithm achieves superior overall performance in optimization quality, algorithmic stability, and computational efficiency. Compared to the SOTA algorithms, SDE improves the best fitness, average fitness, median fitness, standard deviation, convergence quality, and runtime by averages of 2.37%, 15.99%, 13.42%, 80.69%, 66.65% and 39.91%, respectively. Compared to the baseline DE method, SDE achieves even more notable improvements of 11.49%, 17.35%, 17.93%, 34.94%, 72.59%, and 49.29% for the same metrics. Furthermore, AUV field tests verify the feasibility and reliability of the proposed method. Although this preliminary experiment did not account for the underwater terrain and flow field, these factors will be incorporated in future field experiments for further validation.

Dynamic path planning under time-varying conditions remains a challenging yet promising direction for future research. However, the current method is only applicable to environments with static obstacles and does not yet meet rapid responsiveness demands. Encouragingly, the current SDE algorithm already demonstrates significant improvements over other SOTA methods. The abundant sample data generated by SDE presents an opportunity to integrate imitation learning techniques in the future, which could further enhance responsiveness and support a more controllable training process for dynamic environments. Moreover, future work will focus on validating the proposed approach in more diverse and adversarial scenarios, such as extremely narrow passages or highly turbulent regions. These efforts aim to further verify the adaptability and robustness of the SDE algorithm under edge conditions.

Appendix A

For each circle C_i in which the waypoint $\boldsymbol{P}_i$ is located, after determining the set of intersection points $\boldsymbol{P}_{\text{edge},i}$ between C_i and the map boundary L_{edge}, the relative angle of $\boldsymbol{P}_{\text{edge},i}$ with respect to $\boldsymbol{P}_S\boldsymbol{P}_G$ is calculated using Eq. (A1). The polar angles of $\boldsymbol{P}_{\text{edge},i}$ on the left and right sides of $\boldsymbol{P}_S\boldsymbol{P}_G$ that form the smallest absolute angles $\angle\boldsymbol{P}_S\boldsymbol{P}_G\boldsymbol{P}_{\text{edge},i}$ are denoted as $\varphi_{\min,i}$ and $\varphi_{\max,i}$, respectively, as shown in Algorithm A1.

The polar angle φ of any point $\boldsymbol{P}(x,y)$ relative to $\boldsymbol{P}_s$ is

$$\varphi(x,y) = \operatorname{atan2}(y - y_S, x - x_S) \tag{A1}$$

Define the relative angle between $\overrightarrow{\boldsymbol{P}_S\boldsymbol{P}}$ and $\overrightarrow{\boldsymbol{P}_S\boldsymbol{P}_G}$ as

$$\Delta\Phi(x,y) = \begin{cases} \Delta\varphi(x,y), & -\pi < \Delta\varphi(x,y) < \pi \\ \Delta\varphi(x,y) - 2\pi, & \Delta\varphi(x,y) \geqslant \pi \\ \Delta\varphi(x,y) + 2\pi, & \Delta\varphi(x,y) \leqslant -\pi \end{cases} \tag{A2}$$

where $\Delta\varphi(x, y) = \varphi(x, y) - \varphi(x_G, y_G)$. Obviously $-\pi < \Delta\Phi \leqslant \pi$. If $\Delta\Phi > 0$, $\boldsymbol{P}$ is on the right side of $\overrightarrow{\boldsymbol{P}_S\boldsymbol{P}_G}$, otherwise if $\Delta\Phi < 0$, $\boldsymbol{P}$ is on the left side of $\overrightarrow{\boldsymbol{P}_S\boldsymbol{P}_G}$.

Algorithm A1 Calculation method for $\varphi_{\min,i}$ and $\varphi_{\max,i}$

Input: $\boldsymbol{P}_S$, $\boldsymbol{P}_G$, radius $i \cdot r_c$, map boundaries L_{edge};

Output: $\varphi_{\min,i}$, $\varphi_{\max,i}$;

$\varphi_{\min,i} \leftarrow -\pi$; $\varphi_{\max,i} \leftarrow \pi$; $\boldsymbol{P}_{\text{edge},i} \leftarrow \varnothing$; //Initialization

$\boldsymbol{P}_{\text{edge},i} \leftarrow \left\{(x,y) \middle| (x,y) \in C_i \cap L_{\text{edge}}\right\}$;

if $\boldsymbol{P}_{\text{edge},i} \neq \varnothing$ then

  $\boldsymbol{P}_{\text{left},i} \leftarrow \left\{(x,y) \middle| \Phi(x,y) < 0, (x,y) \in \boldsymbol{P}_{\text{edge},i}\right\}$; //Based on Eq. (A2)

  $\boldsymbol{P}_{\text{right},i} \leftarrow \left\{(x,y) \middle| \Phi(x,y) > 0, (x,y) \in \boldsymbol{P}_{\text{edge},i}\right\}$;

  $\boldsymbol{P}_{\min,i} \leftarrow \text{argmax}\left(\Phi(x,y)\right)$, where $(x,y) \in \boldsymbol{P}_{\text{left},i}$;

  $\boldsymbol{P}_{\max,i} \leftarrow \text{argmin}\left(\Phi(x,y)\right)$, where $(x,y) \in \boldsymbol{P}_{\text{right},i}$;

  Converting $\boldsymbol{P}_{\min,i}$ and $\boldsymbol{P}_{\max,i}$ to $\varphi_{\min,i}$ and $\varphi_{\max,i}$ //Based on Eq. (A1)

  if $\varphi_{\max,i} < \varphi_{\min,i}$ then

    $\varphi_{\max,i} \leftarrow \varphi_{\max,i} + 2\pi$;

  end if

end if

return $\varphi_{\min,i}$, $\varphi_{\max,i}$

Funding Supported by the Natural Science Foundation of Heilongjiang Province of China under Grant No. ZD2020E005.

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

References

Bao H, Wang Y, Zhu H, Wang D (2024) Area complete coverage path planning for offshore seabed organisms fishing autonomous underwater vehicle based on improved whale optimization algorithm. IEEE Sensors Journal 24(8): 12887-12903. https://doi.org/10.1109/jsen.2024.3371497

Chang HC, Hsu YL, Hung SS, Ou GR, Wu JR, Hsu C (2021) Autonomous water quality monitoring and water surface cleaning for unmanned surface vehicle. Sensors (Basel) 21(4): 1102. https://doi.org/10.3390/s21041102

Chen G, Shen Y, Qu N, He B (2021) Path planning of AUV during diving process based on behavioral decision-making. Ocean Engineering 234: 109073. https://doi.org/10.1016/j.oceaneng.2021.109073

Chen Z, Yan J, Huang R, Gao Y, Peng X, Yuan W (2024) Path planning for autonomous underwater vehicles (AUVs) considering the influences and constraints of ocean currents. Drones 8(8): 348. https://doi.org/10.3390/drones8080348

Cheng C, Sha Q, He B, Li G (2021) Path planning and obstacle avoidance for AUV: A review. Ocean Engineering 235: 109355. https://doi.org/10.1016/j.oceaneng.2021.109355

Fan J, Qu L (2024) Innovative differential evolution algorithm with double-layer coding for autonomous underwater vehicles path planning in complex environments. Ocean Engineering 303: 117806. https://doi.org/10.1016/j.oceaneng.2024.117806

Fang Z, Wang J, Du J, Hou X, Ren Y, Han Z (2022) Stochastic optimization-aided energy-efficient information collection in internet of underwater things networks. IEEE Internet of Things Journal 9(3): 1775-1789. https://doi.org/10.1109/jiot.2021.3088279

Garau B, Alvarez A, Oliver G (2006) AUV navigation through turbulent ocean environments supported by onboard H-ADCP. Proceedings 2006 IEEE International Conference on Robotics and Automation, Orlando, USA, 3556-3561

Hao K, Zhao J, Li Z, Liu Y, Zhao L (2022) Dynamic path planning of a three-dimensional underwater AUV based on an adaptive genetic algorithm. Ocean Engineering 263: 112421. https://doi.org/10.1016/j.oceaneng.2022.112421

Hu H, Zhang Z, Wang T, Peng X (2024) Underwater glider 3D path planning with adaptive segments and optimal motion parameters based on improved JADE algorithm. Ocean Engineering 299: 117377. https://doi.org/10.1016/j.oceaneng.2024.117377

Hu Y, Wang J, He H, Zhang Y, Cai S, Xie A, Zheng Z (2023) Three-dimensional marine ranching cage inspection path planning integrating the differential evolution and particle swarm optimization algorithms. IEEE Access 11: 109747-109763. https://doi.org/10.1109/access.2023.3321104

Huang Y, Li Y, Zhang Z, Sun Q (2023) A novel path planning approach for AUV based on improved whale optimization algorithm using segment learning and adaptive operator selection. Ocean Engineering 280: 114591. https://doi.org/10.1016/j.oceaneng.2023.114591

Li C, Yao L, Mi C (2024) Fusion algorithm based on improved A* and DWA for USV path planning. Journal of Marine Science and Application 24(1): 224-237. https://doi.org/10.1007/s11804-024-00434-1

Li X, Yu S (2023) Three-dimensional path planning for AUVs in ocean currents environment based on an improved compression factor particle swarm optimization algorithm. Ocean Engineering 280: 114610. https://doi.org/10.1016/j.oceaneng.2023.114610

Liu Z, Ning D, Hou J, Zhang F, Liang G (2024) AUV path planning in a three-dimensional marine environment based on a novel multiple swarm co-evolutionary algorithm. Applied Soft Computing 164: 111933. https://doi.org/10.1016/j.asoc.2024.111933

Luo J, Zhuang J, Jin M, Xu F, Su Y (2024) An energy-efficient path planning method for unmanned surface vehicle in a time-variant maritime environment. Ocean Engineering 301: 117544. https://doi.org/10.1016/j.oceaneng.2024.117544

National Geomatics Center of China (2011) Tianditu. Ministry of Natural Resources of the People's Republic of China. Available from www.tianditu.gov.cn [Accessed on Jun. 9, 2024]

Peng H, Xu Z, Qian J, Dong X, Li W, Wu Z (2023) Evolutionary constrained optimization with hybrid constraint-handling technique. Expert Systems with Applications 211: 118660. https://doi.org/10.1016/j.eswa.2022.118660

Prestero TTJ (2001) Verification of a six-degree of freedom simulation model for the REMUS autonomous underwater vehicle. Master thesis, Massachusetts Institute of Technology, Cambridge, 20-21

Qi Z, Shao Z, Ping YS, Hiot LM, Leong YK (2010) An improved heuristic algorithm for UAV path planning in 3D environment. 2010 Second International Conference on Intelligent Human-Machine Systems and Cybernetics, Nanjing, China, 258-261

Rahimi I, Gandomi AH, Chen F, Mezura-Montes E (2022) A review on constraint handling techniques for population-based algorithms: from single-objective to multi-objective optimization. Archives of Computational Methods in Engineering 30(3): 2181-2209. https://doi.org/10.1007/s11831-022-09859-9

Sahoo SP, Das B, Pati BB, Garcia Marquez FP, Segovia Ramirez I (2023) Hybrid path planning using a bionic-inspired optimization algorithm for autonomous underwater vehicles. Journal of Marine Science and Engineering 11(4): 761. https://doi.org/10.3390/jmse11040761

Storn R, Price K (1997) Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. Journal of Global Optimization 11(4): 341-359. https://doi.org/10.1023/a:1008202821328

Sun B, Niu N (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

Sun Y, Gu R, Chen X, Sun R, Xin L, Bai L (2022) Efficient time-optimal path planning of AUV under the ocean currents based on graph and clustering strategy. Ocean Engineering 259: 111907. https://doi.org/10.1016/j.oceaneng.2022.111907

Sun Y, Zhang L, Zhang G, Cao Jian, Pang Shuo (2025) Development and key technology research of AUVs in water conveyance tunnel detection. Journal of Harbin Engineering University 46(7): 1253-1267. https://doi.org/10.11990/jheu.202404017

Tanabe R, Fukunaga A (2013) Success-history based parameter adaptation for differential evolution. 2013 IEEE Congress on Evolutionary Computation, Cancun, Mexico, 71-78

Wen J, Yang J, Wang T (2021) Path planning for autonomous underwater vehicles under the influence of ocean currents based on a fusion heuristic algorithm. IEEE Transactions on Vehicular Technology 70(9): 8529-8544. https://doi.org/10.1109/tvt.2021.3097203

Xu J, Han Z, Yin L, Yan Z, Yu Y, Ma G (2024) Multi-strategy-based artificial bee colony algorithm for AUV path planning with angle constraints. Ocean Engineering 312: 119155. https://doi.org/10.1016/j.oceaneng.2024.119155

Yan Z, Li Y (2023) Data collection optimization of ocean observation network based on AUV path planning and communication. Ocean Engineering 282: 114912. https://doi.org/10.1016/j.oceaneng.2023.114912

Yan Z, Zhang J, Tang J (2021) Path planning for autonomous underwater vehicle based on an enhanced water wave optimization algorithm. Mathematics and Computers in Simulation 181: 192-241. https://doi.org/10.1016/j.matcom.2020.09.019

Zhai Y, Cui J, Meng F, Xie H, Hou C, Li B (2024) Ship path planning based on sparse A* algorithm. Journal of Marine Science and Application 24(1): 238-248. https://doi.org/10.1007/s11804-024-00430-5

Zhang G, Liu J, Sun Y, Ran X, Chai P (2022) Research on AUV energy saving 3D path planning with mobility constraints. Journal of Marine Science and Engineering 10(6): 821. https://doi.org/10.3390/jmse10060821

Zhang JQ, Sanderson AC (2009) JADE: adaptive differential evolution with optional external archive. IEEE Transactions on Evolutionary Computation 13(5): 945-958. https://doi.org/10.1109/tevc.2009.2014613