IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 31 December 2022, accepted 26 February 2023, date of publication 15 March 2023, date of current version 30 March 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3257025

RESEARCH ARTICLE

Path Planning for Autonomous Underwater Vehicles Based on an Improved Artificial Jellyfish Search Algorithm in Multi-Obstacle Ocean Current Environment

SHUXUAN GUO[1], MINGZHI CHEN[2], AND WEN PANG[2]
[1]Merchant Marine Academy, Shanghai Maritime University, Shanghai 201306, China
[2]School of Mechanical Engineering, University of Shanghai for Science and Technology, Shanghai 200093, China

Corresponding author: Mingzhi Chen (mingzhichen2008@163.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 52001195, Grant 62033009, and Grant 61873161; and in part by the Natural Science Foundation of Shanghai under Grant 20510712300 and Grant 20dz1206700.

ABSTRACT Path planning for autonomous underwater vehicles (AUVs) is a key research focus in the marine domain, requiring consideration of the underwater environment's complexity and the efficiency of the planning algorithms. Firstly, a variety of strategies such as the memory function are integrated into the artificial jellyfish search algorithm (JS) to improve its convergence accuracy, and the improved artificial jellyfish search algorithm (IJS) is obtained. Secondly, this paper establishes a good objective function including the ocean current disturbance model, which helps the IJS algorithm better plan the paths to avoid obstacles and strong side currents. Furthermore, the optimal smoothed paths are obtained by using a cubic spline midpoint interpolation method. Finally, multiple simulation experiments are performed on the multi-obstacle ocean current model with realistic terrain data. The comparison results show that the IJS algorithm with a short running time has the optimal time cost and ocean current penalty cost for the planned path. In addition, the IJS algorithm is also shown to be adaptable in the field of multi-AUV movements.

INDEX TERMS Autonomous underwater vehicle, improved artificial jellyfish algorithm, multi-obstacle ocean current environment, three-dimensional path planning.

I. INTRODUCTION

With the continuous evolution of maritime strategies, underwater unmanned equipment and technology have been valued and developed. In particular, autonomous underwater vehicles (AUVs) have been widely used in civil and military fields because of their advantages of good concealment and strong mobility, which can easily and flexibly carry out underwater operations [1]. It is worth noting that planning a reasonable path to reach the mission point quickly and safely is a prerequisite for AUVs to successfully start underwater operations. Therefore, this paper focuses on the path planning problem of AUVs in a multi-obstacle ocean current environment.

In the 1960s, Lozano-Pérez et al. [2] proposed the idea of path planning, which means that in an environment with obstacles, the motion device needs to find a safe and efficient path from the start point to the target point according to the corresponding evaluation criteria. Nowadays, several researchers have conducted in-depth research on path planning for AUVs [3], [4], [5], mainly including environment modeling and planning algorithms. Environmental modeling is the unified modeling of the collected environmental elements for further construction of the real environment. Of course, the constructed environment model has to be adapted to the path-planning algorithm. There are three main common methods for environment modeling: visibility graph [6], voronoi diagram [7], and grid method [8]. The grid method is easier to understand and operate compared to the other methods, and thus is more popular.

The associate editor coordinating the review of this manuscript and approving it for publication was Nuno M. Garcia.

Commonly used path planning algorithms are mainly divided into traditional algorithms and swarm intelligence optimization algorithms. The traditional algorithms plan a collision-free path by searching and sampling the path nodes. The Dijkstra algorithm is the typical traditional algorithm. In 1996, Arinaga et al. [9] first demonstrated that the Dijkstra algorithm was adapted to path planning for the AUV, which did not take the underwater environment into account. Eichhorn [10] constructed an underwater time-varying environment using weighted directed graphs and successfully planned the optimal path for a single AUV in combination with the Dijkstra algorithm. Later, some researchers made a series of improvements to the Dijkstra algorithm [11], [12] based on the existing ones, which have improved the convergence accuracy of the algorithm. However, the improved algorithms still cannot overcome the drawback of the high running cost and low efficiency, as the Dijkstra algorithm requires a comprehensive search of the path nodes. The A* algorithm and artificial potential field (APF) algorithm are the more popular traditional algorithms. The literature [13] illustrated that the A* algorithm can plan a compliant path for the AUV in complex underwater environments. Nowadays some improvements to the A* algorithm are proposed, and the high research hotspot is the bidirectional A* algorithm [14]. But the A* search must rely on the heuristic values of the adjacent grid of the current path point to determine the next path point, so the A* and its improved algorithms are unable to solve for the optimal path when multiple optimal solutions exist. In 2012, Subramanian et al. [15] used APF for underwater path search and found that it could not fulfill the requirements of AUV path planning. the literature on APF improvement includes [16], [17]. Besides, rapidly exploring random tree (RRT), fast marching (FM), and D* algorithm also belong to the traditional path planning algorithms.

As the complexity of the problem rises, traditional algorithms have obvious difficulties in solving it. Thus swarm intelligent optimization algorithms applied in the field of complex dynamic path planning have become a trend [5]. Due to the ease of operation and fast convergence of particle swarm optimization (PSO), Sun and Liu [18] used this algorithm to solve problems related to large-scale static underwater environments and achieved successful results. The quantum particle swarm algorithm (QPSO) eliminates the moving direction property of particles in the PSO algorithm, which increases the randomness of particles and improves the quality of paths in complex environments [19]. The ant colony (ACO) algorithm relies on pheromones to determine the feasibility of path points, and combines the roulette method to gradually select the location points in the feasible region for planning a reasonable path [20]. This algorithm is suitable for multidimensional underwater environments and has been favored by most researchers. However, it can't be ignored that the ACO algorithm has the fatal drawback of long running time. The wolf pack algorithm (WPA) has low feasibility in achieving multidimensional underwater path planning for an AUV using an excellent leader wolf selection strategy and a "survival of the fittest" update mechanism [21]. In recent years, the marine predator algorithm (MPA) [22] and harris's hawk optimization (HHO) [23] have stood out for their good convergence performance and have been applied in many fields, including path planning. In this work, we introduce the artificial jellyfish search (JS) algorithm to the field of AUV path planning for the first time. In 2021, Chou and Truong [24] proposed the JS algorithm, which achieved an optimal solution to the problem by modeling the behavior of jellyfish searching for food to build a relevant mathematical model. It has been successfully applied to medicine [25] and economics [26]. In general, the common swarm intelligence optimization algorithms have significant advantages over traditional algorithms when dealing with path planning problems. However, they also suffer from common shortcomings: poor stability and robustness, and high dependence on the environment, which need further improvement.

In addition, several new path planning methods have been proposed. Yan et al. [27] proposed an enhanced water wave optimization (WWO) algorithm to plan the optimal path for a single AUV in the two-dimensional (2D) plane, but the method did not apply to the three-dimensional (3D) environment. Han [28] proposed a method adapted to the 3D environment, critical obstacles and surrounding point set (COSPS), which identifies critical obstacles and thus reduces computational complexity. However, this method doesn't take into account the time or distance optimal cost and the smoothness of the path. Wang et al. [29] improved the QPSO algorithm by introducing adaptive parameters, which can successfully identify and avoid obstacles. Che et al. [30] improved the ACO algorithm with heuristic functions and pheromone updates based on the ideas of the PSO algorithm. This improved algorithm has a higher feasibility in the field of path planning compared to the original ACO algorithm. The above two methods have a simulation environment with few obstacles and no ocean current, and their AUV path planning methods are similar to that of the land robot. Although these methods yield better results, they can be biased in practical applications. By improving the neural network, Chen and Zhu [31] proposed a new dynamic neural network model, which helps to capture the local trajectory of the AUV and improved the safety of the path in the ocean current environment. However, the model does not consider the effect of side currents on AUV navigation, which isn't reasonable. The combination of deep reinforcement learning and neural networks is a new hot topic in the field of path planning [32], [33]. This method requires parameter training, which can result in a high computational burden and complex operations.

To improve the above-mentioned problems existing in the field of AUV path planning, the specific works done in this paper are the following.

1) Convergence performance of the basic JS algorithm is improved.
2) A good objective optimization function with the ocean current disturbance model is designed. It gives a

reasonable penalty for the AUV to be disturbed by currents in different directions, thus avoiding side currents and obstacles.

3) The practicality of the simulation path is improved. On the one hand, building an environmental model with realistic terrain data, ocean currents, and external threats to close the gap with the real environment. On the other hand, using the visualization search strategy and cubic spline midpoint interpolation method ensures that the simulation path has multiple scatter points and is smooth.

The remainder of the paper is structured as follows: Section II is large to provide a detailed statement of the problem. Section III presents the designed algorithm model, including the objective function. Section IV performs a variety of simulation experiments and analyses to verify the algorithm model in terms of validity and practicality. The discussion is carried out in the last section.

II. PROBLEM STATEMENT

A. ENVIRONMENTAL MODEL

The development of a good environmental model significantly impacts the timeliness of AUV path planning. The active area of the AUV in this work ranges from 157.1°E to 157.6°E, 19.6°N to 19.9°N, and -6500m below the surface. The realistic ocean terrain data for this area can be obtained by several methods [34] and we represented it using a digital elevation model (DEM). The DEM is usually a data set represented as (X_i, Y_i, Z_i) and Z_i as the elevation of the planar coordinates. In practice, the DEM is easily manipulated and formed into a 2D array, the index of each array containing the elevation of the point Z_i. At the same time, we use the grid method [8] to model the DEM as a grid map. In simple terms, the grid map can be seen as the marine environment of the region cut into a grid of NS*NS*NS, with each grid point represented by (X, Y, Z). When in a real large-scale environment, a single grid generally has a large space. The size of the AUV is relatively small and negligible. Therefore, AUVs are treated as particles in the path planning process. To further represent the harshness of the marine environment, the grid map also includes various shapes of obstacles acting as external threats. Of course, ocean currents can't be ignored and we have applied an ocean current vector to each grid point. The resulting multi-obstacle ocean current environment model is shown in Fig. 1.

B. VISUALIZATION SEARCH STRATEGY

To better solve the AUV path planning problem, we grid the constructed environment, and then give the search strategy for AUV navigation. In this work, the X-axis is taken as the main navigation direction of the AUV. We also perform a non-equal cut Π_i, ($i = 1, \cdots, N$) along the main navigation direction to form N a plane parallel to the YOZ plane. In Π_i, the Y-axis is used as the lateral direction of AUV movement and the Z-axis as the vertical direction, with the AUV as a specific point.

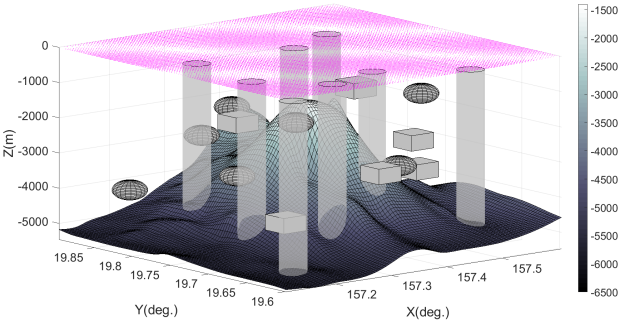


FIGURE 1. The multi-obstacle ocean current environment model (Range: 157.1°*E* to 157.6°*E*, 19.6°*N* to 19.9°*N*, and -6500m below the surface; Gray cylinder: radar reconnaissance range; Sphere, cube: floating obstacles; Pink area: distribution of planar vector ocean currents).

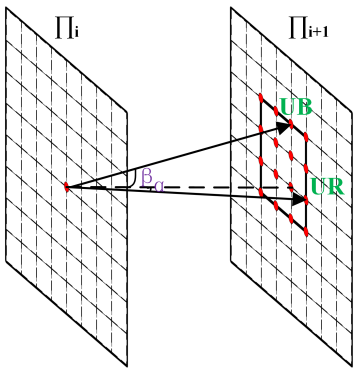


FIGURE 2. Visualization area (Heading angle is α, pitch angle is β, *UB* and *UR* are the maximum sailing distance).

In practice, the AUV has posture (heading angle as α, pitch angle as β, roll angle as 0) limitations that prevent it from navigating excessive distances in the horizontal and vertical directions. Therefore, we set up visualization areas in Π_i, ($i = 1, \cdots, N$). As shown in Fig. 2, when the AUV moves in the main navigation direction to the position point of the Π_i, it can only search within the visualization area of the Π_{i+1} for simplicity and efficiency. The visualization area consists of the maximum lateral sailing distance *UB* of the AUV and the maximum longitudinal sailing distance *UR*. Within the limits of *UB* and *UR*, $\alpha_{\min} \leq \alpha \leq \alpha_{\max}$, $\beta_{\min} \leq \beta \leq \beta_{\max}$. Thus, the visualization search strategy satisfies the constraints of the AUV's posture.

With the above complex environment model and the visual search strategy in place, the 3D path planning problem for the AUV in this work is transformed into a layer-by-layer 2D planar search problem for feasible path points. The problem can be described by a simple schematic diagram in Fig. 3. Firstly, some clarification of Fig. 3 is required: the blue region is the obstacle region. The visualization area contains the red points and the grey points. The grey point is set as an unreachable obstacle point due to its location in the obstacle area. The green points in the plane with Π_1 and Π_{end} are the starting point P_{start} and the target point P_{end} respectively. The purple point is the track point P_i, $i = 1, 2, \ldots, D$. The black

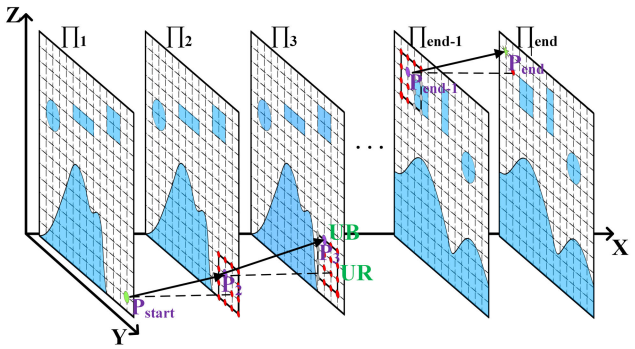


FIGURE 3. AUV path planning diagram.

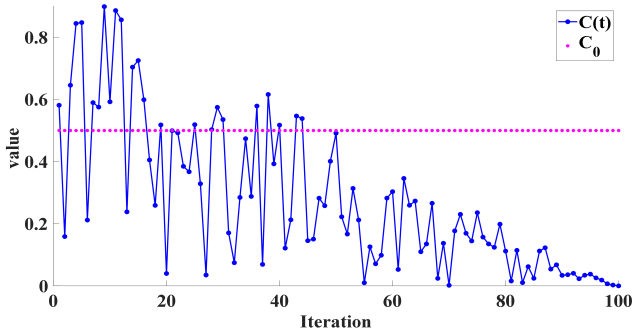


FIGURE 4. Regulating mechanism.

arrow connects the trajectory points P_i in the adjacent plane to form the AUV's navigation path. The problem can therefore be described as the AUV starting from P_{start} and following the trajectory shown by the black arrows to reach P_{end} safely, subject to certain constraints, minimizing the cost function. In this work, the cost function is achieved by optimizing the navigation time. The constraints include that the AUV has to search for the target point and must not collide with obstacles, the AUV posture constraints, and the AUV has to avoid side currents in ocean currents conditions. Simultaneity and conflict avoidance constraints in the case of multi-AUVs co-movement.

III. ALGORITHM MODEL

A. JS ORIGINAL ALGORITHM MODEL

In the original JS algorithm, it is assumed that jellyfish have three main types of foraging behavior: movement following ocean currents, active movement, and passive movement. It is worth noting that jellyfish need to perform three movements through a regulation mechanism. As shown in Fig. 4, this regulation mechanism consists of the regulation function $C(t)$ and the constant $C_0 = 0.5$.

$$C(t) = \left|(2 \cdot rand(0,1) - 1) \cdot \left(1 - \frac{t}{T}\right)\right| \tag{1}$$

where t is the current number of iterations and T is the total number of iterations.

When $C(t) > C_0$ holds, jellyfish will move following the ocean currents. The movement direction of the ocean currents is determined by the optimal jellyfish position in the current population together with the average position of all jellyfish, which can be calculated by (2).

$$\boldsymbol{v_{trend}} = X_g(t) - \lambda \cdot rand(0,1) \cdot X_m(t) \tag{2}$$

$$X_m(t) = \frac{1}{N_p}\sum_{i=1}^{N_p} X_i(t) \tag{3}$$

where $X_g(t)$ is the current optimal jellyfish position, $X_m(t)$ is the average position of all jellyfish, and N_p is the total number of individuals of all jellyfish. λ is the distribution factor with a fixed value of 3. Finally, the jellyfish positions are updated as shown in (4).

$$X_i(t+1) = X_i(t) + rand(0,1) \cdot \boldsymbol{v_{trend}} \tag{4}$$

When $C(t) \le C_0$ and $1 - C(t) < rand(0,1)$ hold, the jellyfish enter the active movement phase. During this phase, information is exchanged between the jellyfish, generating a direction of movement $\boldsymbol{v_{step}}$ as shown in (5), which enables the jellyfish to update their position.

$$\boldsymbol{v_{step}} = \begin{cases} X_i(t) - X_j(t), & if\ X_i(t)\ inferior\ to\ X_j(t) \\ X_j(t) - X_i(t), & otherwise \end{cases} \tag{5}$$

$$X_i(t+1) = X_i(t) + rand(0,1) \cdot \boldsymbol{v_{step}} \tag{6}$$

where $X_j(t)$ is a randomly selected location for the jellyfish j. It can be seen that when the jellyfish i is in a position with a worse food source than the jellyfish j, the jellyfish i will tend to move towards the position of the jellyfish j. Conversely, the jellyfish i will tend to move away from the jellyfish j.

When $C(t) \le C_0$ and $1 - C(t) \ge rand(0,1)$ are established, the jellyfish will take a passive motion in the jellyfish colony. The jellyfish will move around itself in search of a food source. The corresponding updated position of the jellyfish is shown in (7).

$$X_i(t+1) = X_i(t) + rand(0,1) \cdot \delta \cdot (ub - lb) \tag{7}$$

where the coefficient of motion $\delta = 0.1$. ub , lb are the upper and lower bounds of the search space where the jellyfish is located.

The JS algorithm also differs from other algorithms for population initialization and boundary conditions. In the literature [24], the JS algorithm uses the logistic chaos mapping and the opposite bound method to deal with the two cases mentioned above.

B. IMPROVED JS ALGORITHM MODEL AND ANALYSIS

1) IJS ALGORITHM MODEL

In multidimensional complex problems, the JS algorithm suffers from slow convergence and tends to fall into local optimality. To improve the above problems, this work proposes an improved JS (IJS) algorithm as a path planning algorithm for the AUV.

The movement following ocean currents introduces a memory function that is used to re-model the movement of the currents. In single-objective problems, the average position of the population reflects the position distribution of all individuals because there are few constraints and the position of individuals is more evenly distributed. Therefore in solving such problems, the original algorithm uses the direction of the currents fitted by (2) to be favorable for all individuals.

However, in multi-objective problems such as path planning, multiple local extrema are generated in space as the constraints increase. At this point, the applicability of the original algorithm for fitting ocean currents is lower and the following two poor situations arise. One situation is that the average position of the population will be influenced by the extreme positions of some jellyfish and will not accurately reflect the distribution of the positions of all jellyfish. As a result, some jellyfish will blindly follow the movement of the ocean currents and will not have access to better food sources. Alternatively, the ocean currents have a single direction of movement and point to the current global optimum position. At the same time, the step size of the ocean current is adjusted by the rand function, which is too random and causes the algorithm to fall into a local optimum with a high probability. Therefore, this work introduces a memory function: the historical optimal position of the jellyfish $X_p(t)$ is added to generate a valid current $\boldsymbol{v_{cur}}$ as shown in (8).

$$\boldsymbol{v_{cur}} = r_1 \cdot \left(X_g(t) - X_i(t)\right) + r_2 \cdot \left(X_p(t) - X_i(t)\right) \quad (8)$$

$$r_1 = r_2 = 1.5 \cdot rand\,(0, 1) \quad (9)$$

The principle is shown in Fig. 5. The motion vector of the jellyfish from the current position X_i to the global optimum position X_g is $\boldsymbol{v_{ig}}$. The motion vector of the jellyfish from X_i to X_p is $\boldsymbol{v_{ip}}$. We use the adaptation factors r_1, r_2 to adjust $\boldsymbol{v_{ig}}$ and $\boldsymbol{v_{ip}}$ to produce ocean currents with the right direction of motion and step size $\boldsymbol{v_{cur}}$. On the one hand, the fitted ocean currents are favorable for each jellyfish and can drive the jellyfish to find better food sources. On the other hand, the direction and step length of the ocean current is effectively modified by r_1 , r_2, reducing the likelihood of jellyfish becoming trapped in a local optimum. Ultimately, during the new following ocean currents phase, jellyfish update their position according to (10).

$$X_i(t+1) = X_i(t) + \boldsymbol{v_{cur}} \quad (10)$$

The passive movement phase introduces a hybrid strategy that drives the jellyfish in all directions. Analysis of the equations shows that the passive motion of the jellyfish in the original algorithm is directed toward the upper boundary and that the motion steps are highly random. This situation causes the jellyfish to move towards the upper boundary and not in all directions, which is a limitation. Therefore, this work implements a hybrid strategy.

The hybrid strategy consists of a restricted global motion (satisfying the condition: $rand\,(0, 1) \geq 0.5$) and a local following motion, as shown in (11), at the bottom of the next page. On the one hand, the jellyfish, in the course of its restricted global motion, generates a vector of random motions that can reach any position globally. And a linear convergence factor RL is used to limit the motion step size, prompting the jellyfish to actively give up some poor positions during the iterative process and gradually narrow down the global search range with the current position as the core, to achieve the purpose of motion around itself. On the other hand, the jellyfish carries out a local following strategy. Local high-precision development at position $\left(X_g(t) + X_p(t)\right)/2$. Thus, the hybrid strategy will further the ability of algorithmic surveying and exploitation.

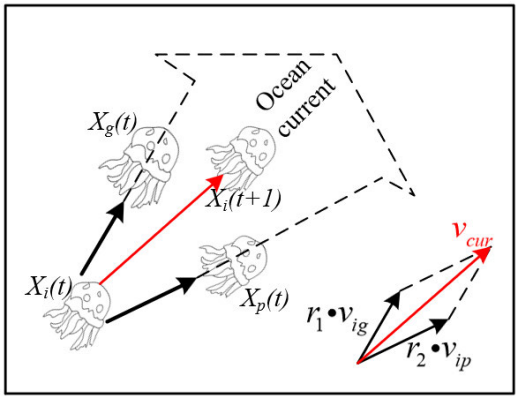


FIGURE 5. Improving the movement of ocean currents.

$$RL = (1 - t/T) \quad (12)$$

$$X_i(t+1) = X_i(t) + \boldsymbol{v_{ran}} \quad (13)$$

where T is the total number of iterations and ub , lb are the upper and lower bounds of the search space respectively.

A hierarchical optimization system is used. The basic JS algorithm uses a conditioning mechanism to perform transitions between different motions. As shown in Fig.4, with the number of iterations increasing, the jellyfish gradually converge to a single motion, which tends to lead the algorithm to a local optimum. To further demonstrate the effectiveness of the improved strategy and to enhance the convergence of the algorithm, this work replaces the regulation mechanism with a hierarchical optimization system. Specifically, the jellyfish populations are grouped in order of fitness from best to worst. The top 1/3 of jellyfish constitute the elite group. Due to their superior position, jellyfish in the elite group perform the passive movement shown in (13). in an attempt to find new food sources locally and globally. The last 1/3 of the jellyfish form the Followers group and will follow the new ocean currents, relying on (10) detached poor food sources. The intermediate jellyfish form the learning group and will enhance inter-individual learning and communication according to (6).

For the treatment of population initialization, the IJS algorithm is in line with other path planning algorithms, using

a simple random initialization. The boundary condition for each path point of a jellyfish satisfies two conditions: it must be within the visualization area as well as within the boundary of the 3D environment.

$$\begin{cases} \left|X^y_{i(j+1)} - X^y_{ij}\right| \le UB \\ \left|X^z_{i(j+1)} - X^z_{ij}\right| \le UR, \end{cases} \quad \begin{cases} X^x_{\min} \le X^x_{ij} \le X^x_{\max} \\ X^y_{\min} \le X^y_{ij} \le X^y_{\max} \\ X^z_{\min} \le X^z_{ij} \le X^z_{\max} \end{cases} \tag{14}$$

2) ALGORITHMIC TIME COMPLEXITY ANALYSIS

The main purpose of time complexity is to measure the speed at which the algorithm runs. Assume a population size as N_P, a dimension as D, and a total number of iterations as T. The time complexity of the IJS algorithm includes initialization, jellyfish position update, and fitness calculation.

Firstly, the IJS algorithm uses a simple random initialization instead of the original logistic chaos mapping, with a time complexity of $O(D * N_P)$. Secondly, for updating the jellyfish position during the iterative process, three improvements are introduced, including the memory function, the hybrid strategy, and the hierarchical optimization system. But these measures are updates to the strategies in the original algorithm and do not add additional time complexity. So the time complexity remains at $O(D * N_P * T)$. Finally, the time complexity of the IJS adaptation calculation is $O(N_P * T)$. Overall the time complexity of the IJS algorithm is $O(D * N_P) + O(D * N_P * T) + O(N_P * T)$, which is the same as the original JS algorithm.

C. OBJECTIVE FUNCTION

The objective function of this path planning module is described as (15), including the time cost function of the path F_t and the good penalty function F_{p_k}, $(k = 1, 2, 3)$.

$$Fit(X) = F_{p_1}(X) \cdot \left(c_1 F_t(X) + c_2 F_{p_2}(X) + c_3 F_{p_3}(X)\right) \tag{15}$$

where $X = \{P_1(P_{start}), P_2, P_3, \ldots, P_D(P_{end})\}$ represents a path consisting of a series of discrete points (both the start and end point). For the weight of the corresponding function c_1, c_2, c_3, set $c_1 + c_2 = 1$, c_3 takes the value 0 or 1. F_{p_1} denotes safety. F_{p_2} is the ocean current disturbance and F_{p_3} is the the success of the path.

1) TIME COST FUNCTION

The time cost function is expressed as follows.

$$F_t(X) = \frac{\varpi_t(X) - \varpi_{t\min}(X)}{\varpi_{t\max}(X) - \varpi_{t\min}(X)} \tag{16}$$

$$\varpi_t(X) = \sum_{i=1}^{D-1} \left(\frac{Dist(P_i, P_{i+1})}{2 * |\boldsymbol{v}_i|} + \frac{Dist(P_i, P_{i+1})}{2 * |\boldsymbol{v}_{i+1}|} \right) \tag{17}$$

$$\varpi_{t\min}(X) = \frac{Dist(P_{start}, P_{end})}{|\boldsymbol{v}_{\max}|} \tag{18}$$

where $\varpi_t(X)$ is the time cost of the path. $Dist(P_i, P_{i+1})$ is the Euclidean distance between the point P_i and the point P_{i+1}, $|\boldsymbol{v}_i|$ and $|\boldsymbol{v}_{i+1}|$ represent the actual scalar velocity of the AUV relative to the ocean current at the points P_i and P_{i+1}, respectively. In general, the AUV navigates with a constant output vector velocity $\boldsymbol{v_{rc}}$. At the same time, the ocean current causes a specific vector velocity $\boldsymbol{v_{ci}}$ at the path point. According to the vector synthesis method, the actual velocity of the AUV's motion is $\boldsymbol{v_i} = \boldsymbol{v_{rc}} + \boldsymbol{v_{ci}}$. $\varpi_{t\min}(X)$is the minimum time cost of the path. $Dist(P_{start}, P_{end})$ represents the shortest distance from the starting point to the target point and $\boldsymbol{v_{max}}$ represents the maximum speed of the AUV actual motion. Conversely, $\varpi_{t\max}(X)$ represents the maximum time cost of the path. Therefore $F_t(X)$ is the minimum time cost of the path in [0, 1]. The smaller the value $F_t(X)$, the lower the time cost of the path.

2) SAFETY FUNCTION

The safety of the path $F_{p_1}(X)$ is described as (19). Due to the presence of marine terrain and external threats in the environmental model, safety is calculated using two methods $J_{r1}(X)$, $J_{r2}(X)$.

$$F_{p_1}(X) = J_{r1}(X) \cdot J_{r2}(X) \tag{19}$$

In a grid map, a grid is an obstacle zone whenever marine terrain is present, and this grid is not passable. Specifically, we binarise the grid points to determine the obstacle zone by applying 0 and 1 as shown in (20).

$$S(x, y, z) = \begin{cases} 0, & if\ Z_m(x, y) \ge Z_g(x, y) + d_{safe} \\ 1, & else \end{cases} \tag{20}$$

$Z_m(x, y)$, $Z_g(x, y)$ represents the elevation of the mountain and the elevation of the grid point located at the point respectively and d_{safe} is the safety distance. If $Z_m(x, y) \ge Z_g(x, y) + d_{safe}$ holds, this grid is an obstacle zone and cannot be reached by the AUV, at which point the value assigned to the grid point is 0. Thus, the safety of AUV passing over marine terrain can be expressed by (21). When an AUV enters an obstacle area, it will be penalized with a fixed value of 10, which ensures that the AUV's path points are away from the obstacle.

$$J_{r1}(X) = \begin{cases} 1, & S(x, y, z) = 1 \\ 10, & S(x, y, z) = 0 \end{cases} \tag{21}$$

External threats include enemy radar threats, local electromagnetic interference threats, floating obstacle threats, etc. Due to the randomness of the shape, size, and location of

$$\boldsymbol{v_{ran}} = \begin{cases} RL \cdot (((ub - lb) \cdot rand(0, 1) + lb) - X_i(t)), & if\ rand(0, 1) \ge 0.5 \\ rand(0, 1) \cdot \left(\left(X_g(t) + X_p(t)\right)/2 - X_i(t)\right), & else \end{cases} \tag{11}$$

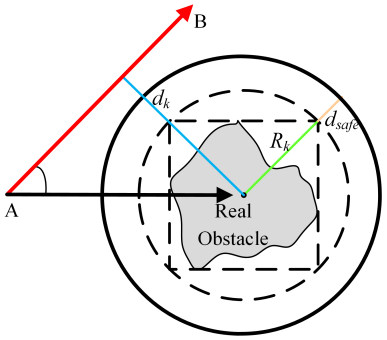


FIGURE 6. AUV obstacle avoidance illustration(R_k: the radius of the obstacle; Red arrow AB: the navigation path of the AUV; d_k: the shortest vertical distance from the center point to the red arrow; d_{safe}: the set safe distance).

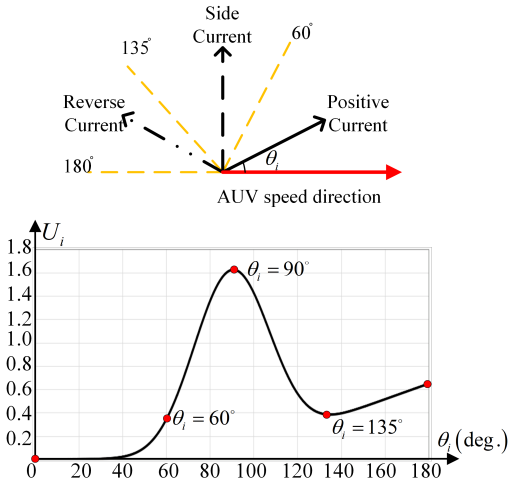


FIGURE 7. Illustration of the penalty for the ocean current (Above is an illustration of ocean current types; Below is the curve of the penalty function for the ocean current).

the external threats, the continued use of the above methods to binarise their location is cumbersome and inaccurate. In this work, the external threats are transformed into different shapes of obstacles and the obstacle areas are modeled. Of course, the real external threat is included in the created obstacles. Due to the long radar survey range, we assume that AUVs can only go around and not pass overhead. The radar threat is treated as an obstacle in the shape of a cylinder as shown in Fig. 1. For all other external threats, we transform them into floating obstacles in the shape of spheres or cubes. As shown in Fig. 6, the grey area is assumed to be a real obstacle in 3D and R_k is the radius of the obstacle. The red arrow AB is the navigation path of the AUV. d_k indicates the shortest vertical distance from the center point of the obstacle to the arrow AB. d_{safe} is the set safe distance. The AUV can only navigate safely if the condition $d_k > R_k + d_{safe}$ is met.

$$J_{r2}(X) = \begin{cases} 1, & if\, d_k > R_k + d_{safe} \\ 10, & else \end{cases} \tag{22}$$

3) OCEAN CURRENT DISTURBANCE FUNCTION

In the ocean current environment model, currents in different directions can exert different forces on the AUV, affecting its safe navigation. In particular, side currents cause greater losses as the AUV cannot effectively resist the impact of side currents. We design the ocean current disturbance function F_{p_2} as shown in (23), which promotes the AUV to use the ocean currents effectively while avoiding the side currents.

$$F_{p_2}(X) = \frac{\sum_{i=1}^{D-1} U_i}{(D-1) * \max(U)} \tag{23}$$

$$U = 1.5 \cdot sin(\mu)^{11.15} + 1.0 \cdot (sin(\mu/3))^3,\ \mu = \frac{\theta}{180} \cdot \pi \tag{24}$$

U is the penalty function for ocean currents and U_i is the penalty value for currents at the time i. μ denotes the conversion between the angular system and the radial system. θ is the angle between the direction of the AUV velocity and the direction of the current velocity.$F_{p_2}(X)$ is in [0, 1]. The smaller the value, the more successful the AUV is in avoiding side currents.

In conjunction with Fig. 7, we illustrate the penalties for the different ocean currents. The ocean current is divided into 3 types: Type 1 is positive ocean current and satisfies the condition as $\theta \in [0°, 60°)$. The ocean current is positive and effective for the AUV movement and suffers the least penalty. Type 2 is reverse ocean current, which satisfies the condition as $\theta \in (135°, 180°]$. Most reverse ocean currents cause the AUV to move slowly. Therefore, the penalty for reverse ocean currents is greater than for positive ocean currents and increases as the angle θ increases. Type 3 is the side ocean current, which satisfies the condition as $\theta \in [60°, 135°]$. Its penalty is based on the following two considerations: a). The penalty value for most side ocean currents is greater than for reverse ocean currents. b). Because the vertical ocean current ($\theta \in [89°, 91°]$) in the reverse ocean current has the highest resistance to AUV navigation, the highest penalty value is set for the vertical ocean currents. Overall, the design of the ocean current penalty function U is reasonable.

$F_{p_3}(X)$ is a larger target penalty value and is only used if the AUV does not reach the target point. The AUV path planning is considered successful only if the AUV searches for the target point in the set visualization area, provided that the other constraints are satisfied. Otherwise, the AUV does not reach the target point and is subject to a fixed penalty value: $F_{p_3}(X) = 5$.

D. B-SPLINE CURVE FOR SMOOTH

Many of the path curves generated using the swarm intelligent optimization algorithms are zigzag and do not match the path trajectory of the AUV in the real marine environment. This work incorporates the B-spline curve [35], which has

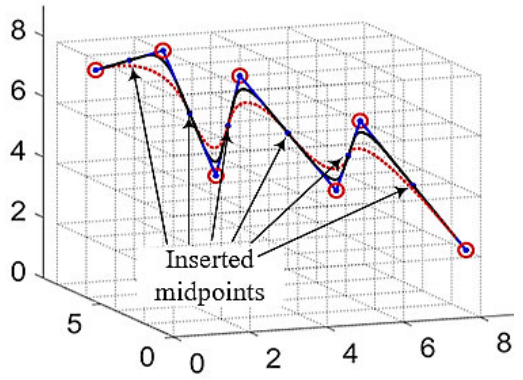


FIGURE 8. Midpoint insertion to improve path following(Black line: the curve with midpoint insertion; Red line: the curve without midpoint insertion).

the continuity of second-order derivatives, to capture the generated path points and fit continuous smooth curves.

The parametric equations of the B-spline curve are as follows.

$$Sp(u) = \sum_{i=1}^{n} \phi_i \cdot B_{i,k}(u) \tag{25}$$

$$B_{i,0}(u) = \begin{cases} 1, & u_i < u < u_{i+1} \\ 0, & others \end{cases}$$

$$B_{i,k}(u) = \frac{u - u_i}{u_{i+k} - u_i} \cdot B_{ik-1}(u) + \frac{u_{i+n+1} - u}{u_{i+k+1} - u_{i+1}} \cdot B_{i+1k-1}(u) \tag{26}$$

where, ϕ_i represents the path points of the AUV. The B-spline basis function $B_{i,k}(u)$ is set to order 3, i.e. $k = 3$. is a node vector, typically taken in [0, 1].

Of course, during this curve fitting correction process, the AUV still needs to perform obstacle avoidance behavior for intersecting complex obstacles. Fortunately, the B-spline can act as a local correction. Secondly, we introduced the midpoint insertion method [36]. As shown in Fig 8, combined with the midpoint insertion method, the B-spline curve can better follow the original line segment to adjust the AUV's obstacle avoidance path.

E. PSEUDO-CODE FOR PATH PLANNING BASED ON THE IJS ALGORITHM

The IJS algorithm for the AUV Path Planning is summarized in Algorithm 1.

Algorithm 1 Path Planning Based on the IJS Algorithm

Input: Environmental model, start point, target point, the extent of visualization area, number of AUVs, and velocity range.

Output: Optimal fitness and optimal smoothing path.

Initialize: (1) number of jellyfish N_p, dimension D, total number of iterations T. (2) The populations were randomly initialized to obtain group initialization paths (which did not necessarily satisfy the AUV condition constraints): $X_i = \{P_{istart}, P_{i2}, \ldots, P_{ik}, \ldots, P_{iend}(P_{iD})\}, i = 1, 2, \ldots N_p$. All jellyfish were used to calculate their respective fitness values by (15).

For t from 1 to T

All jellyfish are listed from lowest to highest fitness value.

For i from 1 to N_p

If $i \leq 1/3$

Update of jellyfish position using (13).

ElseIf $i \leq 2/3$

Update of jellyfish position using (6).

Else

Update of jellyfish position using (10).

End If

Calculate the fitness value $f(X_i)$.

Update the best individual fitness value $X_{bi} = X_i$, *if* $f(X_i) \leq f(X_{bi})$

Update the global best fitness value $X_g = X_{bi}$, *if* $f(X_{bi}) \leq f(X_g)$

End For

Return $X_g, f(X_g)$

The scatter points in the output X_g are used as control points for the optimal path. Use the B-spline curve model to fit a correction to these scatter points and output the optimal smooth path.

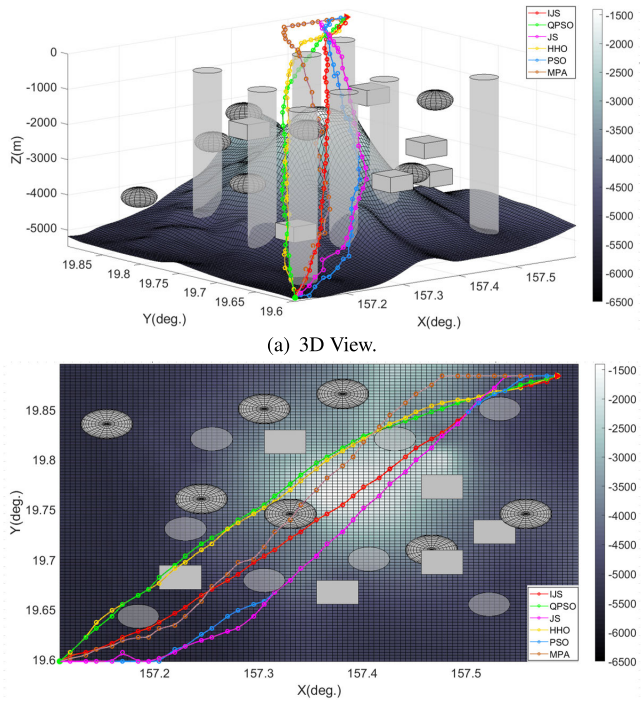
End For

IV. EXPERIMENTS AND ANALYSIS

In this section, we will perform a series of experiments including single AUV path planning in a multi-obstacle static ocean environment and a multi-obstacle ocean current environment, and multi-AUV convergence path planning. The grid map size for the ocean environment is 100*100*100. In the real environment, assuming that 0.1 degree in latitude corresponds to a distance of 11.1 km, 0.1 degree in longitude is $11.1 * cos(Lat)$ km. The IJS algorithm, JS algorithm [24], QPSO algorithm [19], PSO algorithm [18], HHO algorithm [23], and MPA algorithm [22] are selected for comparison experiments. The algorithms to be compared were implemented on MATLAB R2018b simulation software. Computer configuration: Intel(R) Core(TM) i7-10870H, 16 GB RAM, 2.40 GHz main frequency, 64-bit operating system. In addition, all algorithms were initialized uniformly at random. The population size is 45, the dimensions are 40, and $T = 500$, and each algorithm was run 20 times independently.

A. MULTI-OBSTACLE STATIC OCEAN ENVIRONMENT

As currents are not considered in the static ocean environment, the weight parameter $c_2 = 0$ is set in the objective function to remove the penalty for current disturbance. Also $c_1 = 1$. The shortest journey cost path is the same as the



(a) 3D View.

(b) Vertical View.

FIGURE 9. Optimal path planning results in a multi-obstacle static ocean environment.

optimal time cost path. The speed $\boldsymbol{v_{rc}}$ of the AUV in the static ocean environment is set to 3 knots (1 knot is 0.514m/s). The optimal path planning results are shown in Fig. 9 with the different colors corresponding to the trajectories planned by the different algorithms. The AUV starts from the green circle (157.1083°E, 19.5980°N, −5349.05 *meters*) and sails to the destination in the red triangle (157.5881°E, 19.8841°N, −110.29 *meters*). The shortest distance from the starting point to the destination is 40789.30 meters, and the shortest time cost is 26452.21 s.

We tabulate the simulation results of successful solutions from the random independent experiments in Table 1. T_{Best}, T_{Mean}, and STD denote the optimal time cost, the average time cost, and the standard variance of all successful paths. In addition, this work generates AUV paths randomly in the initialization phase, its security cannot be guaranteed and requires continuous iteration through intelligent optimization algorithms to generate feasible paths. However, each algorithm has different execution capabilities, and we use the success rate ($P_{Success}$) to reflect the efficiency and adaptability of the algorithm. $P_{Success}$ refers to the ratio of the number of times each algorithm successfully generates a path that meets the requirements over a set number of experiments. The algorithm's single run time (T_{Single}) is also taken into account.

Analysis with Fig. 9 and Table 1: the algorithms MPA and PSO planning have the worst optimal time cost of 29622.87 s and 29537.65 s. The yellow and green curves in Fig. 9 represent the optimal paths planned by the algorithms HHO and QPSO. The two path curves are similar, with T_{Best} being 27369.97 s and 27107.05 s. The JS algorithm ranks fourth in terms of optimal time cost, with 28236.75 s. The red curve is the smooth path planned by the IJS algorithm, which has T_{Best} = 26741.73 s, closest to the minimum time cost from the starting point to the target point. In terms of average time cost, the IJS algorithm converges the planning time to 27243.69 s, ranking first among all the compared algorithms. The stability of the algorithms can be reflected by the STD evaluation metric. the QPSO algorithm has the best stability with STD = 127.49. the IJS algorithm is second only to the QPSO algorithm with a value of 281.83, which has a greater advantage over other algorithms. The success rate of the HHO, MPA, and PSO algorithms is below 60%. Although QPSO has a high convergence accuracy, its success rate ($P_{Success}$ = 65.00%) is poor and it has low adaptability in the complex and multi-obstacle environment. the JS algorithm ranks second in terms of success rate, but it also has a 25% probability of failing to plan a reasonable path. The IJS algorithm has a success rate of 100% and is suitable for AUV path planning. For the T_{Single} evaluation metric, Except for the HHO algorithm which has a long single run time of 87.84s, the rest of the algorithms have a run time of around 52s. This is due to the increased time complexity of the HHO algorithm, which requires multiple adaptations in the loop structure. Overall, the IJS algorithm has high convergence for solving the AUV path planning problem in the multi-obstacle static ocean environment.

To reflect the stochastic nature of the experiment, we set different starting and ending points for the path planning of the AUV. The simulation results are shown in Fig. 10-12. the HHO and QPSO algorithms converge better, but they have low success rates. the MPA and PSO algorithms have the worst time cost for planning. the MPA algorithm cannot even plan a reasonable path in one experiment. It is clear that in each experiment the IJS algorithm dominates in all evaluation metrics and plans the optimal path that meets the requirements.

B. MULTI-OBSTACLE OCEAN CURRENT ENVIRONMENT

The continuous and complex changes in the velocity vector of ocean currents due to temperature variations, topography, and geostrophic deflection forces severely affect the safe navigation of AUVs. Therefore, robust modeling of ocean currents throughout the ocean environment is essential for AUV path planning. There is a large number of literature that examines the modeling of ocean currents, with mathematical function fitting being the most popular method.

In this work, we construct a time-varying ocean current model using the stream function in [31] as follows.

$$\xi(x, y, t) = 1 - \tanh\left(\frac{y - \alpha(t)\cos(\kappa(x - \varepsilon t))}{\sqrt{1 + (\eta\alpha(t)\sin(\kappa(x - \varepsilon t)))^2}}\right) \quad (27)$$

$$\alpha(t) = \alpha_0 - \beta\cos(\omega_0 t + \gamma) \quad (28)$$

TABLE 1. Results of simulation experiment I in the multi-obstacle static ocean environment (the starting point: (157.1083°*E*, 19.5980°*N*, −5349.05 *meters*), the target point: (157.5881°*E*, 19.8841°*N*, −110.29 *meters*)).

Algorithms	T_{Best} (s)	T_{Mean} (s)	STD (s)	$P_{Success}$	T_{Single} (s)
IJS	**26741.73**	**27243.69**	281.27	**100.00%**	51.90
JS	28236.75	28763.32	369.12	75.00%	52.45
QPSO	27107.05	27278.21	**127.49**	65.00%	51.77
PSO	29537.65	30127.55	534.59	60.00%	52.41
HHO	27369.97	28413.51	657.77	50.00%	87.84
MPA	29622.87	30891.96	827.48	55.00%	**51.64**

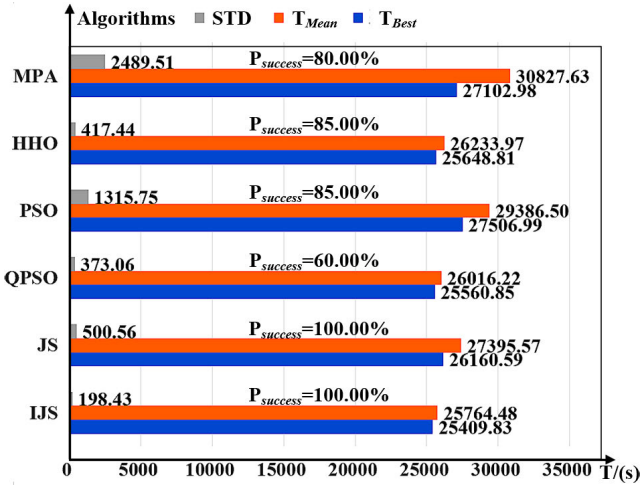


FIGURE 10. Results of simulation experiment II in the multi-obstacle static ocean environment (The starting point: (157.1184°*E*, 19.8901°*N*, −5073.33 *meters*), The target point: (157.5830°*E*, 19.6191°*N*, −551.45 *meters*) The shortest distance is 38915.19 meters. The shortest time is 25236.83 s).

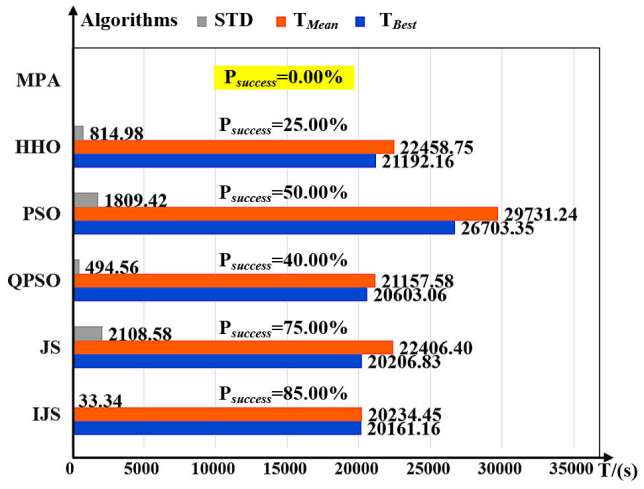


FIGURE 11. Results of simulation experiment III in the multi-obstacle static ocean environment (The starting point: (157.1336°*E*, 19.6552°*N*, −1102.90 *meters*), The target point: (157.5780°*E*, 19.8359°*N*, −3308.69 *meters*) The shortest distance is 30757.05 meters. The shortest time is 19946.21 s).

where the parameters are set: $\kappa = 1$, $\varepsilon = 0.12$, $\eta = 0.84$, $\alpha_0 = 0.12$, $\beta = 0.3$, $\gamma = \pi/2$. The velocity of horizontal currents is $V_c = \left(-\frac{\partial \xi}{\partial y}, \frac{\partial \xi}{\partial x}\right)$.

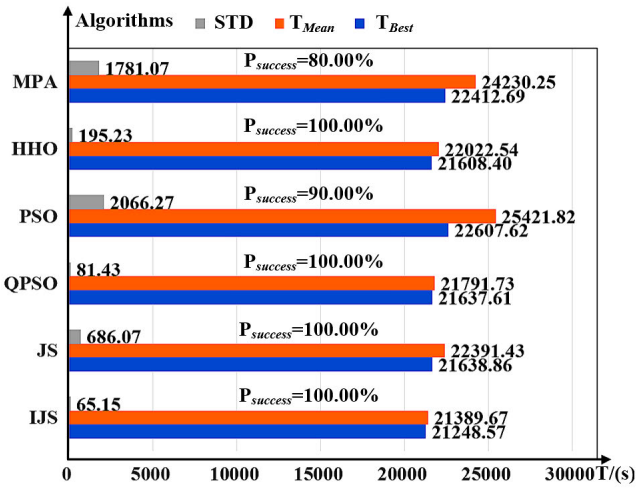


FIGURE 12. Results of simulation experiment IV in the multi-obstacle static ocean environment (The starting point: (157.1790°*E*, 19.8509°*N*, −827.17 *meters*), The target point: (157.5578°*E*, 19.6251°*N*, −4411.59 *meters*) The shortest distance is 32135.91 meters. The shortest time is 20840.41 s).

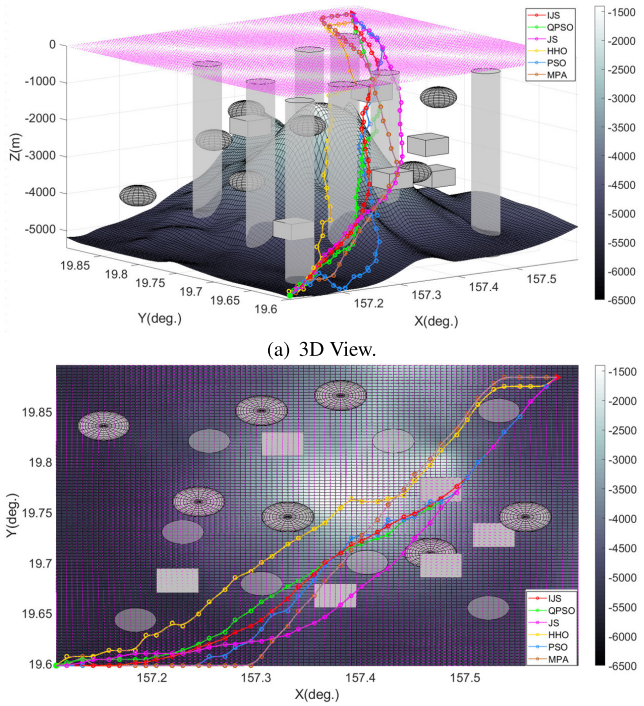
Compared to the strength of the horizontal currents, the vertical currents are less intense and have little impact on the path planning of the AUV, therefore the vertical currents are ignored in this work.

The above time-varying ocean currents are added to the multi-obstacle static ocean environment model to form a dynamic model of ocean currents. After several experiments, the algorithm converges best when $c_1 = 0.7$,$c_2 = 0.3$ are used. The optimal path planning results in the multi-obstacle current environment are shown in Fig. 13, with the starting point (157.1083°E, 19.5980°N, -5349.05 meters) and the target point (157.5881°E, 19.8841°N, -110.29 meters). And the AUV scalar velocity is twice as fast as the scalar velocity of the ocean current ($|\boldsymbol{v_{ci}}| = |\boldsymbol{v_{rc}}|/2$). The yellow and blue curves are the optimal time-cost paths for the PSO and HHO algorithms. It can be seen that the curvature of the yellow and blue curves is too large for the path of the AUV in realistic conditions due to the ocean currents. The other algorithms produce smooth curves.

The simulation results for all Successful programs are tabulated in Table 2. The paths planned by the various algorithms are affected by the ocean currents and their time costs increase compared to the no-current case. Fortunately, the

TABLE 2. Statistics of simulation results in the multi-obstacle ocean current environment.

Algorithms	T_{Best} (s)	T_{Mean} (s)	STD (s)	$P_{Success}$	T_{Single} (s)
IJS	**27982.83**	**29347.53**	**840.24**	**100.00%**	**54.62**
JS	30137.38	33629.18	2931.88	70.00%	54.73
QPSO	28190.70	39278.23	6394.63	65.00%	56.93
PSO	30240.62	37151.21	5520.58	60.00%	54.85
HHO	36513.56	42980.60	4481.69	27.00%	100.32
MPA	36817.99	47163.61	8205.61	50.00%	55.91



(a) 3D View.

(b) Vertical View.

FIGURE 13. Optimal path planning results in a multi-obstacle current environment.

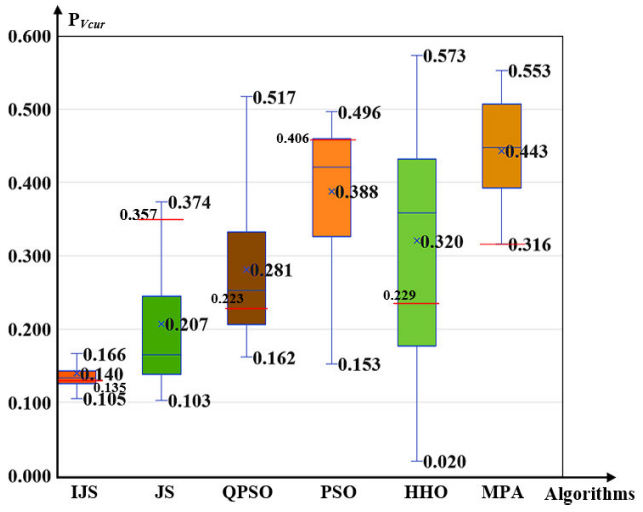


FIGURE 14. Ocean current penalty cost results statistics (The ocean current's scalar velocity is 0.5 times the AUV scalar velocity).

IJS algorithm still shows strong convergence, with a 100% success rate and an optimal time cost of 27982.83 s. All the other algorithms have an optimal time cost greater than 28000.00 s. For the evaluation index T_{Mean}, the algorithms HHO and MPA have the worst costs of 42980.60 s and 47163.61 s respectively. The next are JS, QPSO, and PSO algorithms with time cost in the interval [30000, 40000]. The average time cost of the IJS algorithm is less than 30000 s and the convergence effect is obvious. In addition, the IJS algorithm remains highly stable with STD = 840.24. It is noteworthy that in several independent experiments, all algorithms except the IJS algorithm planned non-conforming paths. This confirms the advantage of the IJS algorithm in path planning in the case of ocean currents as well as its high adaptability. As shown in Fig. 14, we use box plots to represent the current penalty cost P_{Vcur} ($P_{Vcur} \in [0, 1]$), which is calculated by the ocean current disturbance function F_{p_2}. Each box plot is labeled with the maximum value, the average value (the value corresponding to the symbol "C–"), the minimum value, and the current penalty cost (the value represented by the red line) for the optimal time-cost path planned by each algorithm. MPA's average ocean current penalty cost is the highest, at 0.443. HHO can plan the path with the lowest ocean current penalty cost, but the average time cost is higher than 40,000 s. And the large standard variance of the penalty values shows the instability of the HHO algorithm. The other algorithms also plan paths with high P_{Vcur}. The IJS algorithm has P_{Vcur} in the interval [0.105, 0.166] with small fluctuations, indicating that the AUV can successfully avoid side currents and effectively navigate with ocean currents with the IJS algorithm.

Simulation experiments were carried out in strong ocean currents ($|\boldsymbol{v_{ci}}| = 0.8 \cdot |\boldsymbol{v_{rc}}|$, $|\boldsymbol{v_{ci}}| = 1.2 \cdot |\boldsymbol{v_{rc}}|$) to verify the adaptability of the IJS algorithm in strong ocean currents. The statistical results are presented in Table 3 and Fig. 15-16. In the case of $|\boldsymbol{v_{ci}}| = 0.8 \cdot |\boldsymbol{v_{rc}}|$, the MPA algorithm has an increased time cost and is less resistant to ocean current disturbances. All other algorithms can plan a lower optimal time cost path with the help of favorable ocean currents. However, the deterioration of both T_{Mean} and STD indicates that the average time cost and instability of the algorithm increase in more complex cases. Fortunately, the IJS algorithm converges significantly, reaching $T_{Best} = 26142.71$ s $T_{Mean} = 29813.04$ s and Psuccess = 100%. In the case of $|\boldsymbol{v_{ci}}| = 1.2 \cdot |\boldsymbol{v_{rc}}|$, the HHO algorithm and the MPA

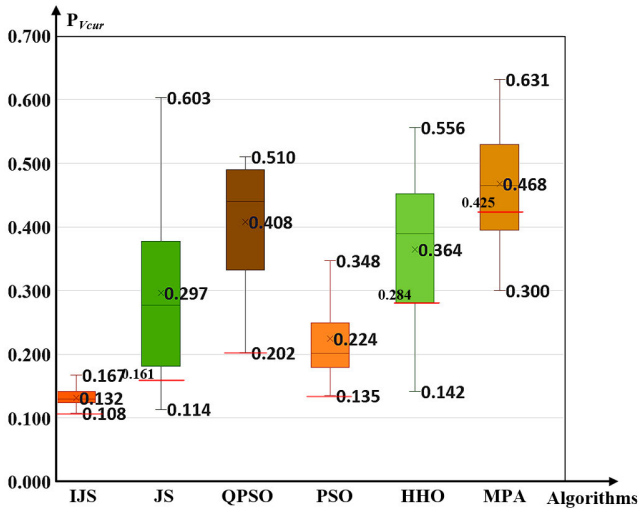


FIGURE 15. Strong ocean current penalty cost results statistics (The ocean current's scalar velocity is 0.8 times the AUV scalar velocity).

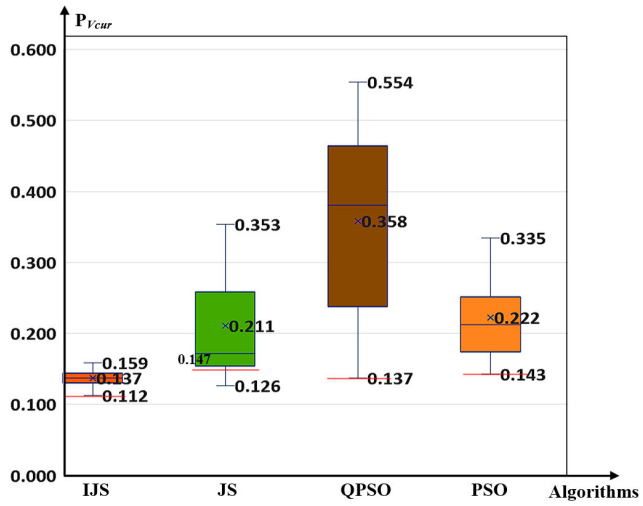
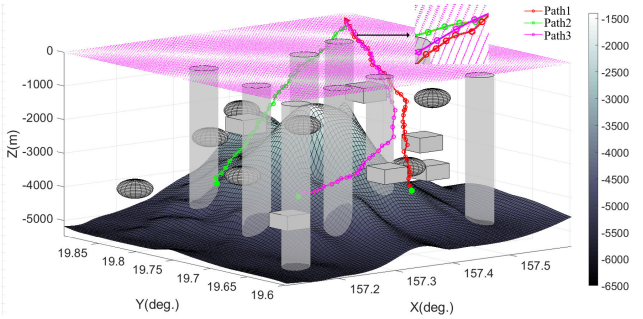


FIGURE 16. Strong ocean current penalty cost results statistics (The ocean current's scalar velocity is 1.2 times the AUV scalar velocity).

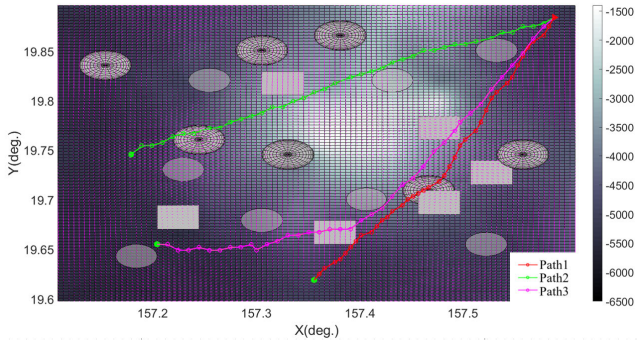
algorithm have failed to plan the path. The average time cost of the algorithms JS, PSO, and QPSO are all higher than 38,000 s, which is at least 10,000 s higher than the IJS algorithm. The IJS algorithm can still plan paths to navigate using strong ocean currents. For the current penalty cost, the IJS algorithm has a significant advantage. the IJS algorithm has the lowest ocean current penalty cost, plans more paths to avoid side ocean currents, and is more resistant to ocean current disturbance.

C. MULTI-AUV CONVERGENCE MODEL

As the difficulty of underwater tasks increases, there is a new trend towards multi-robot co-movement. We model the convergence of multiple AUVs. The model requires that each AUV not only satisfies safety, side ocean current avoidance, and success, but also that multiple AUVs have the same



(a) 3D View.



(b) Vertical View.

FIGURE 17. Multi-AUV optimal path planning results (Path 1: (the starting point: (157.3558°*E*, 119.6191°*N*, −3308.69 *meters*), the target point: (157.5881°*E*, 19.8841°*N*, −110.29 *meters*)); Path 2: (the starting point: (157.1790°*E*, 19.7455°*N*, −3308.69 *meters*), the target point: (157.5881°*E*, 19.8841°*N*, −110.29 *meters*)); Path 3: (the starting point: (157.2043°*E*, 19.6552°*N*, −3308.69 *meters*), the target point: (157.5881°*E*, 19.8841°*N*, −110.29 *meters*))).

time cost (simultaneity) and avoid collisions with each other. In this problem, AUVs have a range of speeds. Simultaneity requires that multiple AUVs start at the same time from their respective starting points, adjust their speed, and reach the next planned path point concurrently. The final result is that multiple AUVs arrive at the target point together to carry out their missions. To avoid collisions between multiple AUVs during navigation, we require multiple AUVs to be located at different path points at the same time, ensuring the safety of the AUVs. The minimum speed of the AUVs is set to 0.5 knots and the maximum speed to 4 knots. The number of AUVs is 3. $|\boldsymbol{v_{ci}}| = 0.2 \cdot |\boldsymbol{v_{rc}}|$.

In the multi-AUV convergence model, the ocean current disturbance function F_{p_2} of the IJS algorithm plays a key role, which can help multiple AUVs travel in combination with ocean currents to meet the requirements of simultaneity and finally reach the target point safely. As shown in Fig 17, the IJS algorithm plans three smooth paths from different starting points to the same target point in a multi-obstacle ocean current environment. path 3 has the longest journey, but with the lowest current penalty cost. This is because the AUV 3 needs to travel in combination with the ocean current under the simultaneity requirement. three paths satisfy the simultaneity with an optimal time cost of 29071.70 s. Twenty independent experiments can achieve a success rate of 65%.

TABLE 3. Statistics of simulation results under strong ocean current environment.

Algorithms	$\lvert\overrightarrow{v}_{ci}\rvert$	T_{Best} (s)	T_{Mean} (s)	STD (s)	$P_{Success}$
IJS	$0.8\lvert\overrightarrow{v}_{rc}\rvert$	**26142.71**	**29813.04**	**1392.14**	**100.00%**
	$1.2\lvert\overrightarrow{v}_{rc}\rvert$	**24565.41**	**28427.09**	**1330.71**	**100.00%**
JS	$0.8\lvert\overrightarrow{v}_{rc}\rvert$	27271.98	41802.74	17181.77	70.00%
	$1.2\lvert\overrightarrow{v}_{rc}\rvert$	25569.06	47805.70	16260.19	50.00%
QPSO	$0.8\lvert\overrightarrow{v}_{rc}\rvert$	27039.31	51941.12	11951.05	60.00%
	$1.2\lvert\overrightarrow{v}_{rc}\rvert$	25074.04	75846.74	54190.94	40.00%
PSO	$0.8\lvert\overrightarrow{v}_{rc}\rvert$	27251.74	32234.78	4660.70	55.00%
	$1.2\lvert\overrightarrow{v}_{rc}\rvert$	26282.91	38740.45	16676.17	60.00%
HHO	$0.8\lvert\overrightarrow{v}_{rc}\rvert$	32225.29	54922.90	18141.11	30.00%
	$1.2\lvert\overrightarrow{v}_{rc}\rvert$	/	/	/	0
MPA	$0.8\lvert\overrightarrow{v}_{rc}\rvert$	48019.19	63591.34	19655.03	35.00%
	$1.2\lvert\overrightarrow{v}_{rc}\rvert$	/	/	/	0

The total cost of the ocean current penalty is 0.41, indicating that the three AUVs mostly navigate in the positive current and reverse current areas. Thus the IJS algorithm is adapted to the multi-AUV movement field.

V. CONCLUSION

To solve the path planning problem for AUV, we build a multi-obstacle ocean current environment and propose a planning method adapted to this environment. The environment model is constructed using the grid method and an AUV visualization search strategy is given. The planning method consists mainly of the IJS algorithm and a well-posed objective function. the IJS algorithm is based on the original JS algorithm, incorporating a memory function, a hybrid strategy, and a hierarchical optimization system. These improved strategies increase the convergence accuracy and robustness of the algorithm. The objective function includes time cost, safety, success, and an ocean current disturbance function, which effectively carries out obstacle avoidance and side current avoidance. the IJS algorithm can plan a reasonable path for AUV navigation under this objective function. the IJS algorithm is simulated in a constructed complex environment model together with algorithms JS, PSO, QPSO, HHO, and MPA. The results show that the IJS algorithm has a significant convergence effect and a high success rate of planning a reasonable path, which is suitable for the path planning field. Finally, the path planning model proposed in this paper is applied in the field of multi-AUV motion.

In the future study, This work needs to be further tested in a real environment with the real AUV. We will consider the actual energy loss, size, and specific attitude changes of the AUV to improve the economy and concealment of AUVs in the path planning process, which is one of our key research directions. In addition, the multi-AUV coordination movement is a hot topic, which is conducive to improving work efficiency. Therefore, it is also crucial to strengthen the practicability of the IJS algorithm in the field of multi-AUV coordinated motion.

REFERENCES

[1] A. Sahoo, S. K. Dwivedy, and P. S. Robi, "Advancements in the field of autonomous underwater vehicle," *Ocean Eng.*, vol. 181, pp. 145–160, Jun. 2019.

[2] T. Lozano-Pérez and M. A. Wesley, "An algorithm for planning collision-free paths among polyhedral obstacles," *Commun. ACM*, vol. 22, no. 10, pp. 560–570, 1979.

[3] F. Bonin-Font, A. Burguera, and J. L. Lisani, "Visual discrimination and large area mapping of *Posidonia Oceanica* using a lightweight AUV," *IEEE Access*, vol. 5, pp. 24479–24494, 2017.

[4] M. Panda, B. Das, B. Subudhi, and B. B. Pati, "A comprehensive review of path planning algorithms for autonomous underwater vehicles," *Int. J. Autom. Comput.*, vol. 17, no. 3, pp. 321–352, Jun. 2020, doi: 10.1007/s11633-019-1204-9.

[5] D. Li, P. Wang, and L. Du, "Path planning technologies for autonomous underwater vehicles—A review," *IEEE Access*, vol. 7, pp. 9745–9768, 2019, doi: 10.1109/ACCESS.2018.2888617.

[6] O. Takahashi and R. J. Schilling, "Motion planning in a plane using generalized Voronoi diagrams," *IEEE Trans. Robot. Autom.*, vol. 5, no. 2, pp. 143–150, Apr. 1989.

[7] B. Oommen, S. Iyengar, N. Rao, and R. Kashyap, "Robot navigation in unknown terrains using learned visibility graphs. Part I: The disjoint convex obstacle case," *IEEE J. Robot. Autom.*, vol. RA-3, no. 6, pp. 672–681, Dec. 1987.

[8] X. Cao, D. Zhu, and S. X. Yang, "Multi-AUV target search based on bioinspired neurodynamics model in 3-D underwater environments," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 11, pp. 2364–2374, Nov. 2016.

[9] S. Arinaga, S. Nakajima, H. Okabe, A. Ono, and Y. Kanayama, "A motion planning method for an AUV," in *Proc. Symp. Auto. Underwater Vehicle Technol.*, Monterey, CA, USA, Jun. 1996, pp. 477–484.

[10] M. Eichhorn, "A new concept for an obstacle avoidance system for the AUV 'SLOCUM glider' operation under ice," in *Proc. Oceans*, Bremen, Germany, May 2009, pp. 572–579.

[11] Y. Singh, S. Sharma, R. Sutton, D. Hatton, and A. Khan, "Feasibility study of a constrained Dijkstra approach for optimal path planning of an unmanned surface vehicle in a dynamic maritime environment," in *Proc. ICARSC*, Apr. 2018, pp. 117–122.

[12] H. Zhang and Z. Cheng, "The method based on Dijkstra of three-dimensional path planning," in *Proc. Chin. Autom. Congr. (CAC)*, Nov. 2020, pp. 1698–1701, doi: 10.1109/CAC51589.2020.9326569.

[13] A. R. Anwary, "Comparison of fuzzy BK-product and A* search algorithm for optimal path finding in unsupervised underwater environment," in *Proc. 8th Int. Conf. Appl. Electromagn.*, Houston, TX, USA, Apr./May 2009, pp. 57–63.

[14] P. Wu et al., "Research on mobile robot path planning based on improved A* algorithm," *Comput. Appl. Eng. Educ.*, vol. 55, no. 21, pp. 227–233, 2019.

[15] S. Subramanian, T. George, and A. Thondiyath, "Obstacle avoidance using multi-point potential field approach for an underactuated flat-fish type AUV in dynamic environment," in *Proc. Int. Conf. Intell. Robot., Automat., Manuf.*, Subang Jaya, Malaysia, Nov. 2012, pp. 20–27.

[16] J. Fu, T. Lv, B. Li, Z. Ning, and Y. Chang, "Three-dimensional underwater path planning of submarine considering the real marine environment," *IEEE Access*, vol. 10, pp. 37016–37029, 2022, doi: 10.1109/ACCESS.2022.3164175.

[17] Y. Chen, G. Bai, Y. Zhan, X. Hu, and J. Liu, "Path planning and obstacle avoiding of the USV based on improved ACO-APF hybrid algorithm with adaptive early-warning," *IEEE Access*, vol. 9, pp. 40728–40742, 2021, doi: 10.1109/ACCESS.2021.3062375.
[18] J. Sun and X. Liu, "Path plan of unmanned underwater vehicle using particle swarm optimization," in *Proc. ISRME*. Nanjing, China: Atlantis Press, Jun. 2015, pp. 1764–1767.
[19] Z. Zeng, K. Sammut, L. Lian, F. He, A. Lammas, and Y. Tang, "A comparison of optimization techniques for AUV path planning in environments with ocean currents," *Robot. Auto. Syst.*, vol. 82, pp. 61–72, Aug. 2016.
[20] H. Wang and W. Xiong, "Research on global path planning based on ant colony optimization for AUV," *J. Mar. Sci. Appl.*, vol. 8, no. 1, pp. 58–64, Mar. 2009.
[21] J. Shen, J. Shi, and L. Xiong, "A route planning method for underwater terrain aided positioning based on gray wolf optimization algorithm," in *Proc. 17th Int. Conf. Intell. Data Eng. Automated Learn.*, in Lecture Notes in Computer Science, vol. 9937, Yangzhou, China, Oct. 2016, pp. 126–133.
[22] A. Faramarzi, M. Heidarinejad, S. Mirjalili, and A. H. Gandomi, "Marine predators algorithm: A nature-inspired metaheuristic," *Expert Syst. Appl.*, vol. 152, Aug. 2020, Art. no. 113377.
[23] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, and H. Chen, "Harris hawks optimization: Algorithm and applications," *Future Gener. Comput. Syst.*, vol. 97, pp. 849–872, Aug. 2019.
[24] J.-S. Chou and D.-N. Truong, "A novel metaheuristic optimizer inspired by behavior of jellyfish in ocean," *Appl. Math. Comput.*, vol. 389, no. 2, Jan. 2021, Art. no. 125535.
[25] M. Abdel-Basset, R. Mohamed, M. Abouhawwash, R. K. Chakrabortty, M. J. Ryan, and Y. Nam, "An improved jellyfish algorithm for multilevel thresholding of magnetic resonance brain image segmentations," *Comput., Mater. Continua*, vol. 68, no. 3, pp. 2961–2977, 2021.
[26] H. Rai and H. K. Verma, "Economic load dispatch using jellyfish search optimizer," in *Proc. CSNT*, Jun. 2021, pp. 301–304, doi: 10.1109/CSNT51715.2021.9509624.
[27] Z. Yan, J. Zhang, and J. Tang, "Path planning for autonomous underwater vehicle based on an enhanced water wave optimization algorithm," *Math. Comput. Simul.*, vol. 181, pp. 192–241, Mar. 2021.
[28] J. Han, "An efficient approach to 3D path planning," *Inf. Sci.*, vol. 478, pp. 318–330, Apr. 2019.
[29] L. Wang, L. Liu, J. Qi, and W. Peng, "Improved quantum particle swarm optimization algorithm for offline path planning in AUVs," *IEEE Access*, vol. 8, pp. 143397–143411, 2020, doi: 10.1109/ACCESS.2020.3013953.
[30] G. Che, L. Liu, and Z. Yu, "An improved ant colony optimization algorithm based on particle swarm optimization algorithm for path planning of autonomous underwater vehicle," *J. Ambient Intell. Hum. Comput.*, vol. 11, no. 8, pp. 3349–3354, Aug. 2020.
[31] M. Chen and D. Zhu, "Optimal time-consuming path planning for autonomous underwater vehicles based on a dynamic neural network model in ocean current environments," *IEEE Trans. Veh. Technol.*, vol. 69, no. 12, pp. 14401–14412, Dec. 2020.
[32] M. Xi, J. Yang, J. Wen, H. Liu, Y. Li, and H. H. Song, "Comprehensive ocean information-enabled AUV path planning via reinforcement learning," *IEEE Internet Things J.*, vol. 9, no. 18, pp. 17440–17451, Sep. 2022, doi: 10.1109/JIOT.2022.3155697.
[33] Z. Li, M. Wang, and G. Ma, "Adaptive optimal trajectory tracking control of AUVs based on reinforcement learning," *ISA Trans.*, Dec. 2022, doi: 10.1016/j.isatra.2022.12.003.
[34] W. H. F. Smith and D. T. Sandwell, "Global sea floor topography from satellite altimetry and ship depth soundings," *Science*, vol. 277, no. 5334, pp. 1957–1962, 1997.
[35] M. Peng and W. Meng, "Cooperative obstacle avoidance for multiple UAVs using spline_VO method," *Sensors*, vol. 22, no. 5, p. 1947, Mar. 2022.
[36] M. Elbanhawi, M. Simic, and R. N. Jazar, "Continuous path smoothing for car-like robots using B-spline curves," *J. Intell. Robot. Syst.*, vol. 80, no. 1, pp. 23–56, Dec. 2015.

SHUXUAN GUO was born in Gansu, China. He is currently pursuing the bachelor's degree in marine engineering with Shanghai Maritime University, with a focus on intelligent optimization algorithms and path planning with his supervisor.

MINGZHI CHEN was born in Fujian, China. He received the B.Sc. and M.S. degrees in marine engineering and the Ph.D. degree in power electronics and power transmission from Shanghai Maritime University, in 2009, 2011, and 2019, respectively. He is currently an Associate Professor with the University of Shanghai for Science and Technology. His current research interests include path planning and multi-AUV coordination systems.

WEN PANG was born in Gansu, China. He received the B.Sc. and M.Sc. degrees in marine engineering and the Ph.D. degree in power electronics and power transmission from Shanghai Maritime University, in 2014, 2019, and 2023, respectively. He is currently a Postdoctoral Researcher with the School of Mechanical Engineering, University of Shanghai for Science and Technology. His research interests include the motion control of autonomous underwater vehicles, formation control for multi-AUV systems, and cooperative transportation by multiple AUVs.