

Article

# An Optimal-Path-Planning Method for Unmanned Surface Vehicles Based on a Novel Group Intelligence Algorithm

Shitu Chen <sup>1</sup>, Ling Feng <sup>1</sup>, Xuteng Bao <sup>1,2</sup>, Zhe Jiang <sup>1</sup>, Bowen Xing <sup>1</sup>  and Jingxiang Xu <sup>1,\*</sup> 

<sup>1</sup> College of Engineering Science and Technology, Shanghai Ocean University, Shanghai 201306, China; cst15298666398@icloud.com (S.C.); zjiang@shou.edu.cn (Z.J.)

<sup>2</sup> Fishery Machinery and Instrument Research Institute, Chinese Academy of Fishery Sciences, Shanghai 200092, China

\* Correspondence: jxxu@shou.edu.cn

**Abstract:** Path planning is crucial for unmanned surface vehicles (USVs) to navigate and avoid obstacles efficiently. This study evaluates and contrasts various USV path-planning algorithms, focusing on their effectiveness in dynamic obstacle avoidance, resistance to water currents, and path smoothness. Meanwhile, this research introduces a novel collective intelligence algorithm tailored for two-dimensional environments, integrating dynamic obstacle avoidance and smooth path optimization. The approach tackles the global-path-planning challenge, specifically accounting for moving obstacles and current influences. The algorithm adeptly combines strategies for dynamic obstacle circumvention with an eight-directional current resistance approach, ensuring locally optimal paths that minimize the impact of currents on navigation. Additionally, advanced artificial bee colony algorithms were used during the research process to enhance the method and improve the smoothness of the generated path. Simulation results have verified the superiority of the algorithm in improving the quality of USV path planning. Compared with traditional bee colony algorithms, the improved algorithm increased the length of the optimization path by 8%, shortened the optimization time by 50%, and achieved almost 100% avoidance of dynamic obstacles.

**Keywords:** unmanned surface vehicles; dynamic obstacle; water currents; eight-directional current resistance; path smoothness



**Citation:** Chen, S.; Feng, L.; Bao, X.; Jiang, Z.; Xing, B.; Xu, J. An Optimal-Path-Planning Method for Unmanned Surface Vehicles Based on a Novel Group Intelligence Algorithm. *J. Mar. Sci. Eng.* **2024**, *12*, 477. <https://doi.org/10.3390/jmse12030477>

Academic Editor: Sergei Chernyi

Received: 8 January 2024

Revised: 3 March 2024

Accepted: 3 March 2024

Published: 11 March 2024



**Copyright:** © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

An unmanned surface vehicle (USV) is an autonomous, crewless watercraft capable of self-navigating with high intelligence. Path planning is essential in the USV's navigation system, optimizing its trajectory for enhanced precision, efficiency, and safety. This process involves the algorithmic computation of the best route based on a predefined map and start point. USVs, adept in aquatic and subaqueous operations, surpass human physical limits, boosting underwater task efficiency. Their autonomous features include precise targeting, cyclic engagement, self-destruct capabilities, and swift evacuation, reducing their dependence on human operators and logistical support. This makes path planning crucial in USVs' industrial and scientific applications, garnering significant interest from researchers and industry professionals.

Path planning is divided into two main categories: global and local. Global path planning uses established algorithms, like Dijkstra's A-star, the Probabilistic Roadmap Method (PRM), Rapidly Exploring Random Tree (RRT), and swarm intelligence-based methods, such as ant colony and genetic algorithms. These often represent the environment as a grid map for spatial discretization. Local path planning, in contrast, occurs in real time during the USV's movement, employing methods like the artificial potential field method, Fuzzy Logic, Reinforcement Learning, and the Dynamic Window Approach.

The global planning of the USV involves determining the optimal path from the starting point to the target position in the known surrounding environment. For global path

planning and multi-objective joint optimization path control, a previous paper proposed a method that maximizes the unit time profit in the traveler's problem, establishes a global-path-planning model, and takes the path distance, stability, residence risk, economic cost, and safety as target constraints. A path control model between two task locations was established. The CSPSO algorithm was introduced to solve global path planning [1]. However, a new mechanism was not used to improve the collision avoidance rate in the collision avoidance link. In order to make the unmanned surface vessel better applied in the monitoring fields of inland rivers, reservoirs, coasts, etc., a bidirectional search strategy was proposed to search nodes simultaneously from the starting point and the target point. An improved A-star global-path-planning algorithm was introduced based on the vertical distance node deviation factor of the starting point and target point connection [2]. However, the practical application of unmanned ships does not consider the comprehensive influence of environmental factors, so the reliability requirements for unmanned ships are higher. In order to improve the overall path search efficiency and smoothness, an improved A-star algorithm with a bidirectional alternating search strategy was proposed [3]. However, the improved A-star algorithm is carried out in a static environment. In the whole process, obstacles are fixed and known. Many mobile robot application scenarios are dynamic environments with unknown obstacles. To improve the optimality of global search paths, a forward search optimization (FSO) algorithm was proposed to shorten the paths planned by Dijkstra, A\*, and other global algorithms based on searching. It was combined with the SHPP method of the IFIS and IAPF algorithms to smooth the global path [4]. However, the resulting path cannot ensure optimality in a continuous space, and its smoothness can be further improved. In order to solve the problem that the ant colony algorithm makes it difficult to meet the mission requirements of underwater robots, another paper proposed a global-path-planning algorithm (IQACA) that utilizes the efficiency of quantum computing and the optimization ability of the ant colony algorithm [5]. However, in the algorithm fusion process, the correlation between multiple objectives, the weight of each objective in the cost function, and the actual environmental load's influence on the USV's path optimization process energy consumption were not considered, and the USV's dynamic and kinematic constraints were not added to the cost function.

In USV local path planning, only partial environmental information can be obtained, and prior information, such as obstacle locations, cannot be known in advance. The algorithm deals with the problem of avoiding obstacles during movement by identifying the dynamic conditions of the environment and establishing relationships between various elements. Local path planning is used in dynamic environments, where robots need to constantly sense environmental changes and adjust their paths in a timely manner to avoid collisions. A fuzzy-based path-planning approach was proposed for various uncertain obstacles in unknown, highly uncertain, and dynamic environments to find the locally optimal path for robotic movement [6]. The motion of USVs in complex environments requires multi-objective optimization and multi-modal constraints. An improved fuzzy ant colony algorithm (ACO-FL) was proposed after considering factors like wind, current, waves, and dynamic obstacles to handle local avoidance path planning [7]. However, this study had limitations in modeling obstacles with elliptical trajectory motion and considering water current effects. It did not take into account complex dynamic environments with more moving and static obstacles, or dynamic ocean current speeds. To address the dynamic path-planning problem for environmental monitoring USVs under complex offshore navigation conditions, a hybrid path-planning algorithm combining global and local approaches was proposed based on bi-level programming ideas. An improved particle swarm optimization algorithm (PSO) was proposed based on ocean environmental information for global path planning. Then, an improved artificial potential field (APF) algorithm was used for local dynamic obstacle avoidance based on the globally optimal path to solve the local minimum problem [8]. However, the USV and dynamic obstacles were simplified in the experiment as particles without considering their specific shapes. Path optimization ignored external factors like wind, waves, and current and only con-

sidered geometric constraints in the workspace without considering the USV's kinematic model and constraints. Also, the dynamic obstacle avoidance process did not consider the rules of obstacle avoidance at sea. Particle swarm optimization has also been used for path planning, but its local minimum, premature convergence, and low efficiency hinder its widespread application. An improved localized particle swarm optimization algorithm (ILPSO) was proposed to address the local minimum issue of PSO. By modifying the inertia weight, acceleration factor, and locality, the algorithm's convergence speed was increased, and its ability to search for the locally optimal path was enhanced [9]. However, this algorithm still falls short in complex, large-scale, and three-dimensional environments.

Obstacle avoidance has always been a continuous research focus in path planning. Traditional algorithms are limited to avoiding static obstacles and often ignore the impact of dynamic obstacles that exist in practical applications of USVs. To address the challenges of single global path planning and local path planning, which cannot guarantee the existence of a feasible solution in complex environments, an algorithm was proposed that uses the A\* algorithm for global path planning to generate a global path for the USV to reach the target point, and it introduces an improved DWA algorithm with weight coefficients for sea conditions for local path planning. This algorithm was ultimately verified for its real-time obstacle avoidance capabilities [10]. To address the computational workload required by current algorithms for dynamic path planning, which makes them difficult to use in dynamic situations, and the simplicity of other methods but their tendency to be stuck in local minima, an improved simulated annealing algorithm was proposed that introduces an initial path selection method and deletion operation for dynamic path planning [11], but it did not consider the precise obstacle avoidance of multiple targets in actual application scenarios. To address mobile robots moving in dynamic environments, a predictive path-planning algorithm based on the fast search random tree algorithm was proposed. It searches for an approximate global optimal path in advance and then performs simple predictions of the motion of the robot itself and dynamic obstacles before replanning the path for dynamic obstacles [12]. However, the RRT algorithm's attempt to force pruning can result in significant time and additional computational overhead. To address USVs' inability to efficiently avoid dynamic obstacles, NT\* was introduced to establish a mapping relationship and add minimum path costs to the objective function. The IDWA algorithm was proposed to achieve real-time obstacle avoidance and improve dynamic collision avoidance capabilities [13], but it is difficult to directly establish a universal standard for parameter selection due to the limitations of single scenarios and strong coupling between weight parameters. Due to the complexity of the interactive environment, dynamic obstacle avoidance path planning poses a significant challenge to the mobility of intelligent agents. Dynamic path planning is a complex multi-constraint optimization problem with combinations of constraints. Some existing algorithms can easily get stuck in local optima when avoiding moving obstacles in complex interactive environments, resulting in defects in convergence speed and accuracy. An improved Q-learning algorithm was proposed, which greatly improves the convergence speed and accuracy of the algorithm and finds better paths in dynamic obstacle path planning [14]. However, as the problem size increases, the Q table in Q-learning algorithms expands, increasing the complexity of the algorithm. The Adaptive and Sound Searching Algorithm can only obtain the initial optimal path in a static environment. The Morphin algorithm was introduced to achieve real-time obstacle avoidance for moving obstacles [15], but it does not consider the impact of different shapes and motion states of dynamic obstacles on USV motion. To solve the problem that global path planning cannot avoid dynamic obstacles, an improved APF algorithm was used for local path planning, and a dynamic warning mechanism was adopted to optimize the step size, forming an improved ACO-APF hybrid algorithm [16]. Finally, it can effectively avoid unknown static and dynamic obstacles.

At present, path-planning research in the land and air domains is relatively mature, and various land mobile robots and drones have been able to complete path-planning tasks well. However, USVs typically operate in complex and changing underwater environments,

where the drag force of water is much greater than that in the air or on land. Therefore, path planning for USVs must not only consider timely obstacle avoidance but also take into account the impact of the water flow on USV navigation. The water flow, as an energy flow, has a great impact on the navigation of USVs, which can exhibit both thrust and resistance. To address the issue of dynamic ocean current changes in complex marine environments, an improved MPP framework with a multi-level approach, backbone CA and RC, and self-adjusted path planning was proposed to adapt to time-varying ocean current environments [17]. To solve the problem of two-dimensional autonomous path planning for unmanned underwater vehicles (UUVs) in environments influenced by both ocean currents and obstacles, an improved Fireworks–Ant Colony Hybrid algorithm was proposed to establish a two-dimensional Lamb vortex ocean current environment model with randomly distributed obstacles, in which circular obstacles were equivalent to square grid cells [18], while considering energy consumption, travel time, and distance as reference factors in the cost function. Considering the complexity of the underwater environment and the efficiency of the planning algorithm, an improved artificial jellyfish search algorithm (IJS) was proposed, which integrated memory functions with multiple strategies to establish a target function that included ocean current disturbance models to avoid the impact of obstacles and strong lateral flow on AUV movement [19]. However, the actual energy loss, size, and attitude changes of AUVs were not considered. To address the issue of path planning for autonomous surface vessels not considering external factors such as wind, waves, and currents that affect navigation safety, an adaptive inertia weight was introduced into the fitness function of an improved particle swarm optimization algorithm [20] to increase the resistance to wind and wave forces. This improved path safety but did not consider the actual energy loss of the ship. In response to complex ocean current environments, an underwater path-planning method based on near-policy optimization was proposed by constructing a deep reinforcement network as a decision-making control system (UP4O), which integrated features of obstacles with current state information [21]. It was demonstrated that, under complex three-dimensional ocean currents, limited prior knowledge, and local information, it could find a time-saving collision-free path that narrowed the gap between AUV theoretical research and actual marine applications. The control difficulty and collision risk of unmanned surface vehicles (USVs) were increased due to an insufficient consideration of the impact of ocean environment factors. To address this issue, a new evaluation function was proposed by redesigning the kinematics model for underwater robots and classic practical local-path-planning methods based on dynamic windowing [22], ensuring that underwater robots could adjust to planned paths and avoid obstacles in real time during navigation in real ocean environments. However, changes in the strength of environmental factors still significantly affected the path length and navigation time of USVs. To eliminate the impact of ocean currents on optimal path planning for UUVs, an intelligent algorithm with an adjusted function was proposed [23], but the actual scenario maneuverability was not considered. As winds and waves had a great impact on obstacle avoidance for unmanned aerial vehicles (UAVs) in actual flight environments, a UAV path-planning method based on COLREGS (Convention on the International Regulations for Preventing Collisions at Sea) and an improved DWA algorithm (UDWA) was proposed [24]. By prioritizing speed sampling areas and enhancing speed functions in the target function, the impact of winds and waves on UAVs was transformed into speed changes. However, this method did not consider the impact of ship structure and shape on actual speed. To address path-planning issues for unmanned sailboats in complex marine environments, an integrated algorithm based on improved adaptive ant colony optimization and rolling window methods was proposed [25]. However, it did not consider more complex dynamic environments or real ship application scenarios.

In the path-planning of robots, traditional algorithms demonstrate strong optimization capabilities, but most paths feature many turning points, a high degree of curvature, and a lack of directionality, which ultimately result in significant energy loss and path redundancy for the robots. To address the disadvantages of traditional APF-ACO, such

as multiple turning points and a high degree of curvature, an IAPF-IACO algorithm (Improved Artificial Potential Field-Improved Ant Colony Optimization) that integrates turning point optimization and circular spline fitting was proposed, which significantly reduced the number and length of turning points in the path and obtained smoother and more stable path results [26], but this approach does not consider the impact of ocean environmental factors on submarine navigation, limiting its practical application. Aiming to solve the problem that meta-heuristic methods can easily fall into local minima, a high-order continuous Bezier curve was proposed, with the constraints being the length of the expected path of the robot and the non-collision safety requirement, creating an optimized problem for smooth path planning and transforming the smooth path-planning problem into an optimization problem searching for the control node positions of the Bezier curve [27], but this algorithm can only solve the global planning of static obstacles, and when unknown or dynamic obstacles arise due to geometric topology changes, the control points of the path may also experience sub-optimal or insufficient conditions. Since RRT-related algorithms have problems such as slow convergence speed, too many turning points, and uneven path generation in ship track planning, a new method called PI-DP-RRT was proposed, which combines information from the advanced automatic identification system (AIS) with Douglas–Peucker (DP) compression ship path planning, applying an improved DP algorithm and a new path optimization method to optimize the path to improve its smoothness and practicality [28], but it does not fully consider more complex navigation environments and dynamic obstacle avoidance. To address the issues of slow convergence, the tendency to be easily trapped in local optima, and poor smoothness in the traditional ant colony algorithm in mobile robot path planning, an improved ant colony algorithm with a path smoothness factor was proposed. The algorithm incorporates an updated pheromone update rule, which reduces the number of robot turns. Additionally, a new path evaluation function was introduced to enhance the pheromone discrimination for effective paths [29]. A new method based on a fourth-order Bezier transition curve and an improved particle swarm optimization algorithm was proposed for the smooth path planning of mobile robots [30], but it does not consider SPP through the real-time position of mobile robots; traditional A\* algorithms generate routes that are constrained by the map resolution and may not be compatible with a USV’s non-holonomic constraints. A smooth A\* path-planning algorithm suitable for autonomous navigation control systems of unmanned ships was proposed to improve route performance, reduce unnecessary “sawtooth”, and provide more continuous routes [31], but this algorithm does not consider hydrodynamic effects from the perspective of path planning. To address the issue of repeated paths and many turning points during agricultural robot work, an immune ant colony B-spline interpolation-based path-planning algorithm was proposed to eliminate path repetition and angular turning points, increasing the smoothness of agricultural robot path trajectories [32], but it does not consider the actual energy consumption of robots.

## 2. Overview of Improved Algorithms

This article introduces the Dynamic Obstacle Avoidance Strategy with Smooth Optimization for the Artificial Bee Colony (DOASO-IABC) algorithm, offering an innovative solution to address real-world challenges. The algorithm seamlessly combines dynamic obstacle avoidance strategies and smooth optimization techniques. The main contributions of this work are as follows:

- (1) In scenarios with known static obstacle positions, it leverages the artificial bee colony (ABC) algorithm to determine the global optimal path.
- (2) When confronted with moving obstacles and water current influences, it adapts by integrating dynamic obstacle avoidance strategies and anti-current optimization methods to identify locally optimal paths, significantly enhancing the practicality and effectiveness of path planning.

- (3) To further elevate path quality, this article introduces the integration of the Advanced\_sprcv curve optimization model. This addition refines path smoothness, ultimately augmenting the navigational performance of unmanned surface vehicles.

This article is organized as follows: Section 1 briefly introduces the necessity of path planning for USVs; Section 2 briefly describes the improvement of swarm intelligence algorithms in this article; Section 3 proposes the structure of the USV path-planning algorithm; Section 4 analyzes the results of simulation experiments; Section 5 provides corresponding experimental conclusions.

### 3. Methods

#### 3.1. Environmental Modeling

The working environment of the USV in this paper is set in a two-dimensional space. In the two-dimensional environment, when setting obstacles, the obstacle is expanded to the maximum diameter of the robot. At the same time, the robot is regarded as a point mass to simplify the complexity of the problem under the premise of ensuring safety. The grid method is easy to implement and has a clear and direct path display. Therefore, the grid method is used to model static spatial two-dimensional spaces [33]. The grid method model is shown in Figure 1. The coordinates are set with the upper left as the origin, from left to right as the positive direction of the x-axis, and from top to bottom as the positive direction of the y-axis. Each grid length is marked as the unit length and is set as unit 1. Starting from the upper-left corner, it is marked as serial number 1. The serial numbers increase sequentially from left to right and from top to bottom. The white grid in the figure is the reachable grid, the black grid is the obstacle grid, the yellow grid is the starting point, and the red grid is the endpoint. The mod() function is used for modulo (remainder) operations. Let the grid serial number be c, and let n be the number of grids in each row and column. The ceil() function represents rounding up. The correspondence between serial numbers and coordinates in Figure 1 is shown in Formula (1).

$$\begin{cases} x = \text{mod}(c, n) - 0.5 \\ \text{if } x = -0.5, x = 19.5 \\ y = n + 0.5 - \text{ceil}(c/n) \end{cases} \quad (1)$$

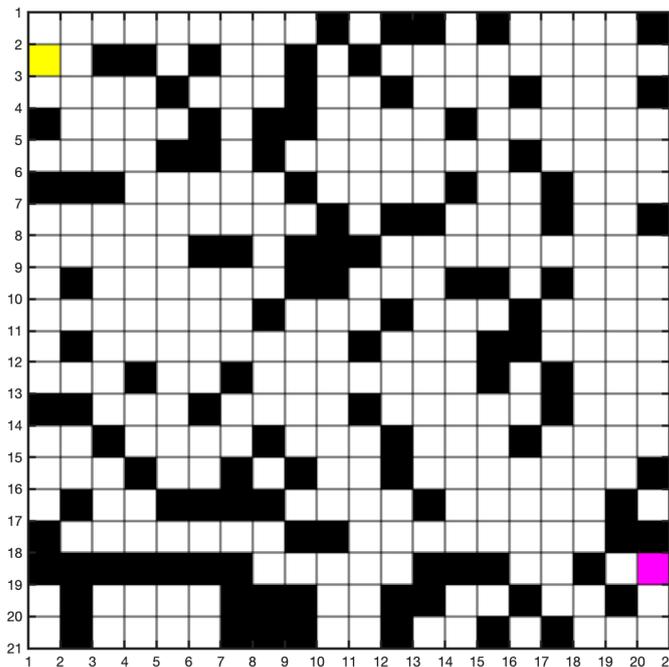


Figure 1. Environmental model of raster method.

### 3.2. Modeling of 2D Path Planning for Underwater Robots

Path planning is the process of finding an optimal route from a starting point to a target point. In the intricate underwater environment, it is crucial to consider not only the robustness of the algorithm but also the actual conditions of the underwater environment, including factors like water flow and dynamic or static obstacles beneath the surface.

#### 3.2.1. Traditional 2D Artificial Bee Colony Algorithm

The bee colony algorithm (ABC) is an optimization technique that mimics the clustering intelligence observed in honey bee foraging behavior. It does not require explicit knowledge of the problem’s specifics; rather, it relies on comparing the relative advantages and disadvantages of the problem. Through the local optimization efforts of individual bees, the global optimal value can be highlighted within the colony, resulting in a faster convergence rate. The overall path-planning process can be conceptualized as bees extracting honey from a map and sharing information about honey source locations. Three distinct types of bee colonies follow distinct foraging rules, ultimately converging on an optimal path that effectively avoids obstacles. The algorithm’s overall flow is summarized in Figure 2 below.

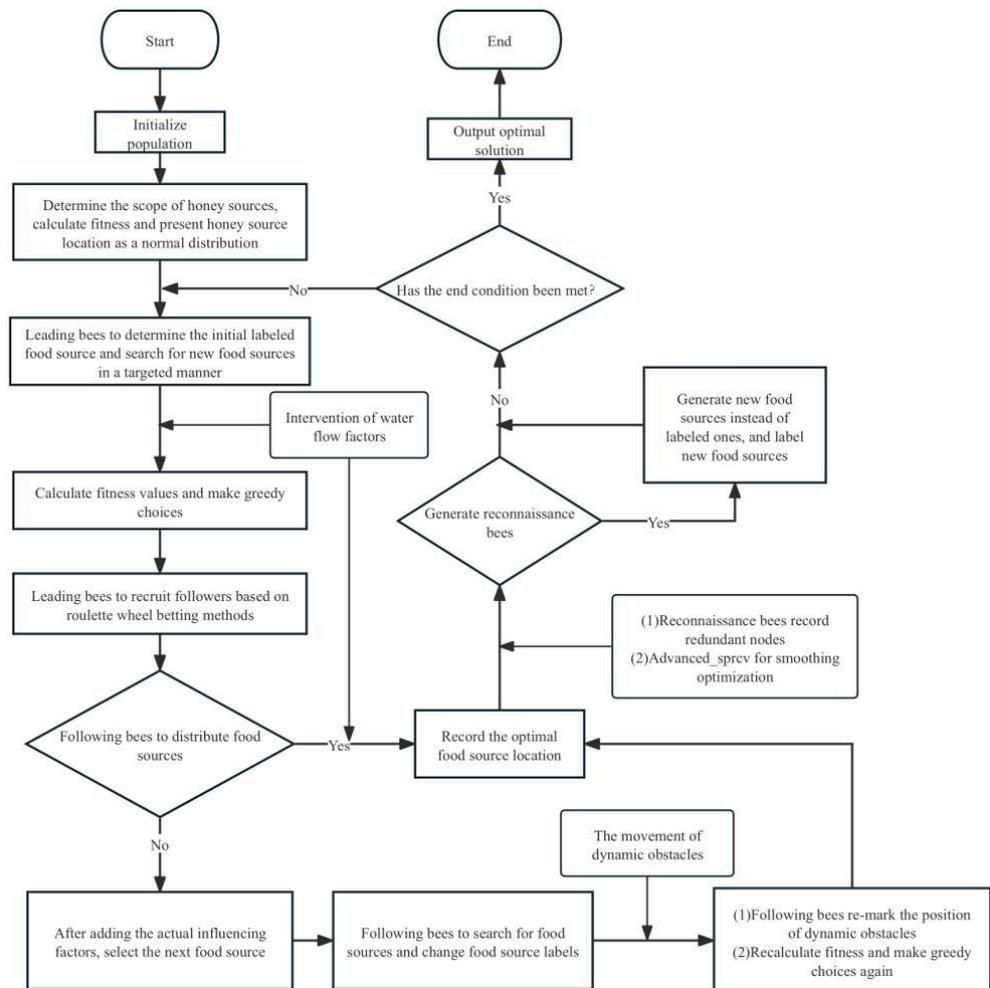


Figure 2. Flowchart of improved bee colony algorithm.

#### (1) Leading Bee Search Mechanism

The size of the bee colony  $num\_polution$ , the maximum number of iterations  $max\_gen$ , and control parameters  $limit$  are initialized; initial nectar sources are generated according

to the random production nectar Formula (2); and the fitness function value  $fit$  of each solution is calculated.

$$x_{id} = L_b + (U_b - L_b)rand(0, 1) \tag{2}$$

The lead bee conducts a neighborhood search in the vicinity of the current nectar source  $i$ , finds the node  $k$  that is not  $j$ , generates new solutions according to the neighboring search Formula (3), and calculates its fitness function.

$$v_{ij} = x_{ij} + \varphi_{ij}(x_{ij} - x_{kj}), i \neq k \tag{3}$$

In this equation,  $k$  is a random number generated and not equal to  $i$ , a random number within  $\varphi_{ij} = [-1, 1]$ ; the neighborhood search formula controls the generation of food sources in the vicinity of position  $x$ , which represents the comparison of bees to adjacent food sources. According to the greedy selection method, the state of the new honey source is determined as retention or replacement. If the fitness value of the new solution is better than that of the old solution, the leading bee updates the old solution to the new one; otherwise, it retains the old solution. Formula (2) represents the greedy selection formula.

$$fit_i(t) = \begin{cases} \frac{1}{1+fit_i(t)} & fit_i(t) \geq 0 \\ 1+|fit_i(t)| & fit_i(t) < 0 \end{cases} \tag{4}$$

In this equation,  $fit_i(t)$  represents the richness of the nectar source corresponding to the fitness value of the  $i$ th solution, and  $f_i(t)$  is the objective function of the  $i$ -th individual for the optimization problem. If the fitness of the new solution is better than that of the old solution, the new solution is regarded as the current optimal solution; otherwise, the nectar exploitation is  $f_i(t) + 1$ .

(2) The probability selection mechanism of following bees

The fitness value of each nectar source is calculated, the following probability  $p_i$  is obtained, and the following bees select nectar sources according to this probability  $p_i$ . Then, they search the neighbors of the selected nectar source according to the position update formula, generate new solutions, and calculate their fitness. Then, they make greedy choices between the current nectar source and the new nectar source and determine the nectar amount in the nectar source. Formula (5) represents the selection probability of a roulette wheel for calculating solutions.

$$p_i = \frac{fit_i}{\sum_{k=1}^{NP} fit_k} \tag{5}$$

(3) The spatial random search strategy of scout bees

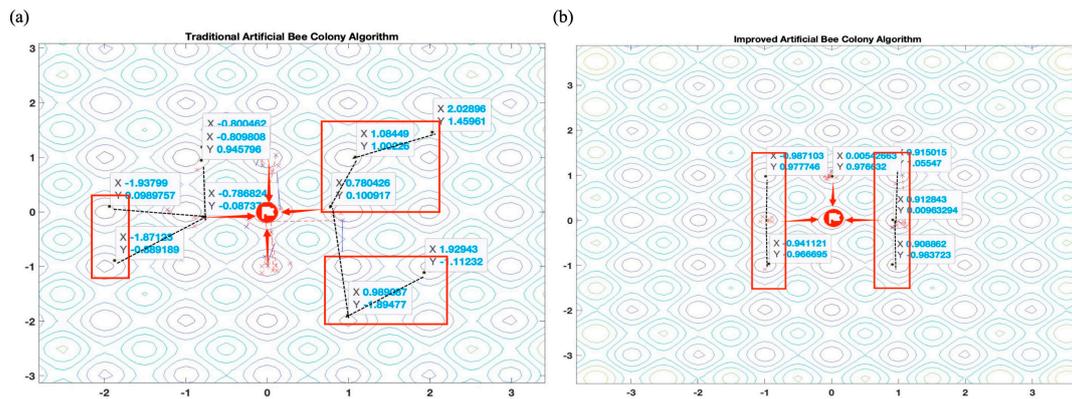
When a nectar source fails to be updated after a limited number of iterations, it will be abandoned, and the nectar source will be recorded in the table. At the same time, according to Formula (6), the following bees corresponding to this nectar source will become scout bees and randomly generate new nectar sources in the global scope to replace the original nectar source (remembering the best nectar source found so far).

$$x_i^j = x_{min}^j + (x_{max}^j - x_{min}^j)rand() \tag{6}$$

In this equation,  $x_i = (x_{i1}, x_{i2}, \dots, x_{id})$  is a  $d$ -dimensional vector,  $x_{max}^j$  and  $x_{min}^j$  represent the upper and lower limits of the  $j$ -th dimension of the problem, respectively, and  $rand()$  is a random number within  $[0,1]$ .

### 3.2.2. Improved Two-Dimensional Artificial Bee Colony Algorithm

The traditional artificial bee colony algorithm has strong global optimization ability; it can adjust parameters adaptively and has a certain tolerance to parameter changes. It also exhibits good robustness in the two-dimensional water environment. In this paper, considering the unpredictable obstacles in the actual water environment and the impact of the water flow on the robot’s motion state, a dynamic obstacle avoidance strategy was integrated into the traditional bee colony algorithm, and a water flow correction function was added to ensure that the USV can still reach the target point position after being impacted by the water flow and smooth the path based on improvement. The red border and the black dashed line in Figure 3 represent the bee colony’s optimization area and the bee colony’s actual movement direction, respectively. The red arrow indicates the final decision of the bee colony in the next optimal position. As shown in Figure 3a, the traditional artificial bee colony algorithm’s bee colony is greatly affected by water flow during the process of searching for the next optimal point. The landing points are relatively random, and the bee colony is unable to decide on the impact of the water flow while finding the target point, resulting in the “lost” state of its motion trajectory. As shown in Figure 3b, under the resistance of the water flow, the movement trajectory of the improved artificial bee colony algorithm shows a relatively “unified” state. Meanwhile, the improved bee colony can reduce the impact of the water flow and find a better landing point after resisting the water flow without randomly appearing landing points.



**Figure 3.** Optimization of bee colony motion using traditional artificial bee colony algorithm and improved artificial bee colony algorithm.

#### (1) Real-time dynamic obstacle avoidance strategy

During the movement of a USV in complex water areas, hardware limitations may prevent the recognition of certain obstacles, and unpredictable obstacle movement information may even arise. Therefore, it is essential to provide timely feedback for unforeseen events during unmanned robot navigation, enabling it to avoid obstacles and reach the target point safely and efficiently. The key to achieving autonomous navigation for a USV also lies in the technology for active obstacle avoidance. To avoid obstacles without prior knowledge, USVs utilize real-time obstacle recognition systems, such as vision sensors or obstacle avoidance sonar, to detect unknown obstacle information. In this paper, without assuming sensor recognition, an improved artificial bee colony algorithm combined with a dynamic obstacle avoidance strategy is employed to achieve the timely evasion of local obstacles.

The fitness function in the traditional bee colony algorithm is given by Equation (4). Assuming that the number of obstacles is  $n$  and their positions are denoted by  $o_1, o_2, \dots, o_n$ , a new fitness function is defined by Equation (7).

$$Fit_i(t) = fit_i(t) + \sum_{i=1}^n \omega_i \cdot d_{obs-USV} \tag{7}$$

in which  $\omega_i$  is the weight related to obstacle  $i$ , and  $d_{obs-USV}$  is the distance between the artificial bee colony  $x$  and obstacle  $i$ .

There may be multiple unknown moving obstacles in an unknown environment, so it is necessary to detect the obstacles in advance. Among them, the obstacles that have no impact on the planning process are removed, then the risk of the remaining obstacles to the overall planning is evaluated, and finally, the next step is planned.

(2) Initial screening of obstacles

First, a virtual straight line from the USV to the target point is established and denoted by  $L_{st}$ , as shown in Equation (8).

$$L_{st} : y = k_{st}x + b_{st}, x \in (x_{USV}, x_{tar}) \tag{8}$$

In the formula,  $k_{st} = \frac{y_{tar}-y_{USV}}{x_{tar}-x_{USV}}$  is the slope of the line  $L_{st}$ ;  $\frac{x_{tar}y_{USV}-x_{USV}y_{tar}}{x_{tar}-x_{USV}}$  is the intercept of the line  $L_{st}$ ; and  $b_{st}$  represents the intercept of the line between the USV and the target point at the intersection of the  $y$ -axis after extension.  $(x_{USV}, y_{USV})$  and  $(x_{tar}, y_{tar})$  are the position coordinates of the USV and the target point, respectively.

The distance between the obstacle and USV is denoted by  $d_{obs-USV}$ , and the distance between the obstacle and the line  $L_{st}$  is denoted by  $d_{obs-st}$ . When  $d_{obs-USV}$  or  $d_{obs-st}$  is less than the corresponding safety threshold, it is judged that the obstacle will affect the subsequent planning of the USV, so obstacles that may affect the planning are screened out at the beginning. The screening method reduces the impact of far obstacles on the trajectory and improves the efficiency of path planning. The expressions  $d_{obs-USV}$  and  $d_{obs-st}$  are given by Formulas (9) and (10).

$$d_{obs-USV} = \sqrt{(x_{obs} - x_{USV})^2 + (y_{obs} - y_{USV})^2} \tag{9}$$

$$d_{obs-st} = \frac{|k_{st} \cdot x_{obs} - y_{obs} + b_{st}|}{\sqrt{k_{st}^2 + 1}} \tag{10}$$

The corresponding safety thresholds of the two are  $d_{obs-USV}^*$  and  $d_{obs-st}^*$ .

When  $d_{obs-USV} < d_{obs-USV}^*$  or  $d_{obs-st} < d_{obs-st}^*$ , the obstacle is involved in the calculation of the planning algorithm.

### 3.2.3. USV's Obstacle Avoidance Strategy for Dynamic Obstacles

(1) USV left- and right-side dynamic obstacle planning

According to the moving direction of the obstacle, a ray starting from the current position of the obstacle and pointing to the moving direction of the obstacle is established [33]. This ray is denoted by  $L_{obs}$ , as shown in Figure 4. Based on the movement trend of the obstacle, a judgment is made as to whether there is a possibility of collision between the USV and the obstacle. The expression of the ray  $L_{obs}$  is as follows:

$$L_{obs} = \begin{cases} y = \tan \theta_{obs}(x - x_{obs}) + y_{obs}, \theta_{obs} \in (-\frac{\pi}{2}, \frac{\pi}{2}) \cup (\frac{\pi}{2}, \frac{3\pi}{2}) \\ \theta_{obs} \in (-\frac{\pi}{2}, \frac{\pi}{2}), x \geq x_{obs}; \\ \theta_{obs} \in (\frac{\pi}{2}, \frac{3\pi}{2}), x \leq x_{obs} \end{cases} \tag{11}$$

In the formula,  $\theta_{obs}$  represents the angle between the obstacle's moving direction and the  $x$ -axis, which is a dynamic variable;  $(x_{obs}, y_{obs})$  represents the obstacle's position coordinates, which change with its movement.

It is assumed that the intersection point of the ray  $L_{obs}$  and the line segment  $L_{st}$  is  $P(x_p, y_p)$ , where the horizontal and vertical coordinates are represented in Formula (12) as follows:

$$\begin{cases} x_p = \frac{\tan \theta_{obs} \cdot x_{obs} - y_{obs} + b_{st}}{\tan \theta_{obs} - k_{st}} \\ y_p = \frac{k_{st} \cdot x_{obs} - y_{obs} + b_{st}}{\tan \theta_{obs} - k_{st}} \cdot \tan \theta_{obs} + y_{obs} \end{cases} \tag{12}$$

In the formula,  $k_{st} = \frac{y_{tar} - y_{USV}}{x_{tar} - x_{USV}}, \frac{x_{tar}y_{USV} - x_{USV}y_{tar}}{x_{tar} - x_{USV}}$ , and  $(x_{obs}, y_{obs})$  represent the position coordinates of obstacles, and  $\theta_{obs}$  indicates the angle between the motion direction of obstacles and x-axis.

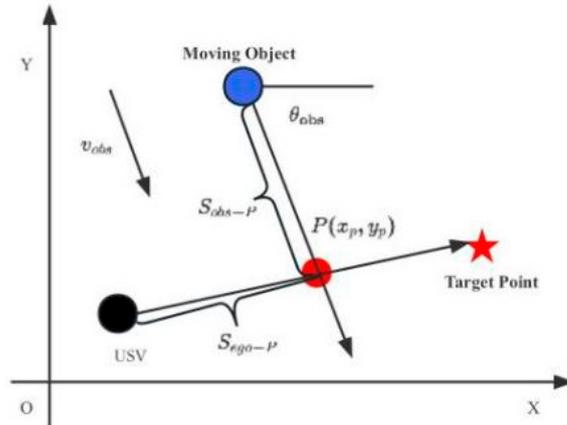


Figure 4. Dynamic obstacle planning diagram on both sides of USV.

Based on the collision point  $P$ , we calculate the time cost of the USV and obstacles from their current positions to the collision point  $P$ . Suppose that the current speed of the USV is  $v_{USV}$  and the acceleration is  $a_{USV}$ , while the current moving speed of obstacles is  $v_{obs}$  and the acceleration is  $a_{obs}$ . Both of them move toward the point  $P$  under the current state. The time costs of the USV and obstacles are  $t_{USV-P}$  and  $t_{obs-P}$ , respectively, which are expressed by Formulas (13)–(16), respectively.

$$t_{USV-P} = \frac{\sqrt{2a_{USV}S_{i-P} + v_{USV}^2} - v_{USV}}{a_{USV}} \tag{13}$$

$$t_{obs-P} = \frac{\sqrt{2a_{obs}S_{obs-P} + v_{obs}^2} - v_{obs}}{a_{obs}} \tag{14}$$

$$S_{obs-P} = \sqrt{(x_{obs} - x_p)^2 + (y_{obs} - y_p)^2} \tag{15}$$

$$S_{USV-P} = \sqrt{(x_{USV} - x_p)^2 + (y_{USV} - y_p)^2} \tag{16}$$

In the formula,  $S_{USV-P}$  represents the distance from the current position of the USV to the intersection  $P$ , and  $S_{obs-P}$  represents the distance from the current position of the obstacle to the intersection  $P$ .

The times at which the USV and obstacle will reach the point  $P$  are compared. If  $t_{obs-P} < t_{USV-P}$ , then the obstacle reaches the intersection  $P$  first:

- (1) When the obstacle is close to the point  $P$  and the distance from the USV to the obstacle is greater than the safe distance, the USV maintains its original state of motion;
- (2) When the distance from the USV to the obstacle is less than the safe distance, the USV decelerates in advance to avoid collision;
- (3) When the obstacle moves away from the point  $P$ , the USV resumes its original speed and moves to the target point.

If  $t_{obs-P} = t_{USV-P}$ , then the USV and the obstacle are in a parallel state.

To ensure the collision-free driving of the USV, a collision priority is set under the premise that the USV maintains a safe distance from obstacles. When the USV is close to obstacles (within a safe distance), deceleration measures are taken. When the distance between the USV and obstacles expands, the USV accelerates to its initial speed (without considering increasing speed).

If  $t_{obs-p} > t_{USV-p}$ , then the USV reaches the intersection  $P$  faster.

If the distance from the USV to the obstacle is less than the safe distance, the USV will judge the safety grid around the obstacle and avoid it. Secondly, when the USV judges that the risk of collision increases, it increases its driving speed to quickly avoid the dangerous influence range of the obstacle.

If the distance from the USV to the obstacle is greater than the safe distance, the USV maintains its original speed and continues to move to the target point.

(2) Obstacle planning on the front side of the USV

When the obstacle is moving in the same direction as the USV on the front side of the USV, the motion state of the obstacle is judged, and the next state of the USV is selected [33], as shown in Figure 5. In this figure, the blue rectangle represents the USV; the black solid circle represents a moving obstacle, whose direction of movement is consistent with that of the USV; the red pentagram represents the target point of the USV; and the white arrow indicates the forward direction of the USV.

- (1)  $v_{USV} = v_{obs}$ , and when the distance  $d_{USV-obs}$  is safe, the USV maintains the original speed.
- (2)  $v_{USV} > v_{obs}$ , and when the distance  $d_{USV-obs}$  is less than the safe distance and there are static obstacles around, it does not meet the obstacle avoidance distance, so the USV stops driving.
- (3)  $v_{USV} > v_{obs}$ , and when the distance  $d_{USV-obs}$  reaches the variable distance and meets the obstacle avoidance distance, the USV accelerates to pass the obstacle.
- (4) Mathematical model of water flow constraint

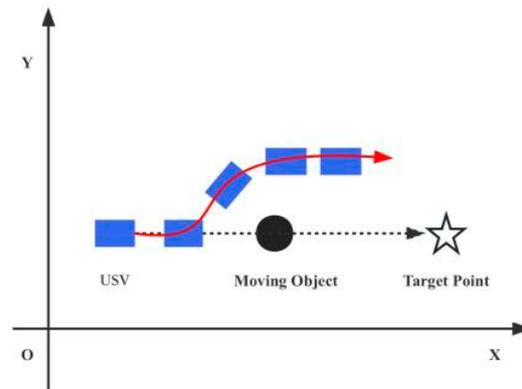


Figure 5. USV's same-direction dynamic obstacle avoidance planning chart.

In the actual motion process of the robot on the water surface, the robot is inevitably affected by the water flow. In a two-dimensional grid map, the general choice is to simulate the water flow in four directions relative to the robot, as shown in Figure 6, where the water flow velocities from the four directions are represented by (clockwise from top to bottom)  $v_{up}, v_{right}, v_{down}, v_{left}$ .

This is not considered in actual scenes, as the actual influence of the water flow is not limited to four directions, so the remaining four positions are considered the actual water flow constraints, as shown in Figure 7, where the water flow velocities from the four new angle directions are represented by (clockwise from top right to top left)  $v_{up\_right}, v_{down\_right}, v_{down\_left}, v_{up\_left}$ .

In view of the different directions of the water flow, it was simplified to a mathematical model. Considering that the water flow impact is a new constraint factor, its effect is similar to that of dynamic obstacles. Dynamic obstacles have a "Repulsion" effect on the USV. The actual effect of the water flow is not only "Repulsion", which acts as an obstacle, but also "Traction", which pushes the USV. Therefore, the schematic diagram of this mathematical

model is shown in Figure 8. There are three forces in the figure, where  $f_{USV}$  represents the direction of the force acting on the USV,  $f_{up\_right}$  represents the force acting on the water flow from the northeast direction, and  $f_{joint}$  represents the combined force of  $f_{USV}$  and  $f_{up\_right}$ .

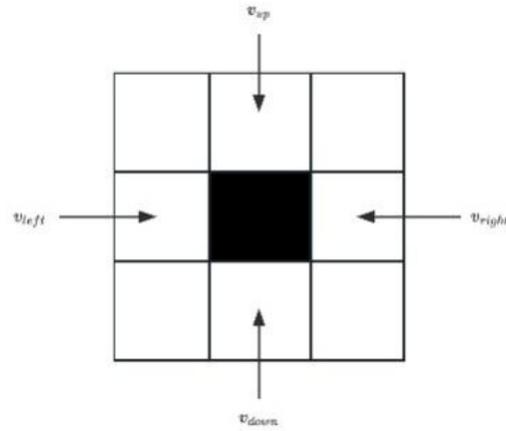


Figure 6. The impact of water flow in all four directions.

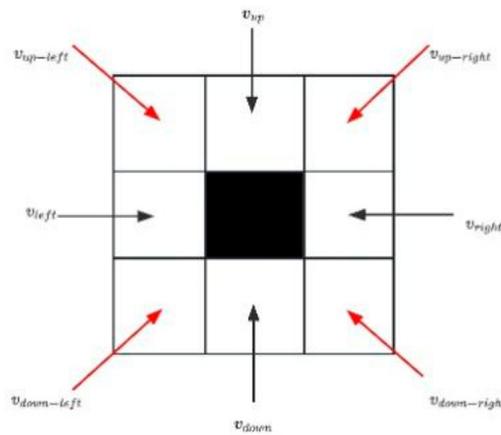


Figure 7. Influence of water flow in all directions.

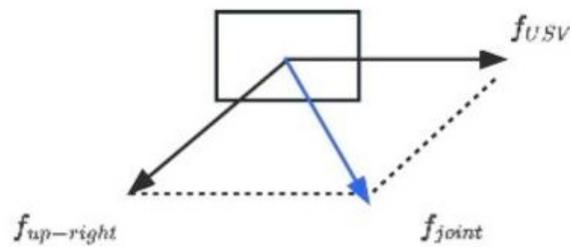


Figure 8. The combined motion direction of the USV after being affected by the water flow.

It can be seen from Figure 8 that the direction of the resultant force acting on the USV is not the desired direction of the USV’s movement in the ideal state. Therefore, a direction adjustment is needed. The direction adjustment constraint formula is shown in Equation (17). Based on Figure 8, a new USV direction model is established, as shown in Figure 9. The figure contains different forces, with  $f_{USVactual}$ , indicated by the red arrow, indicating the actual forward force of the USV after two combined forces.

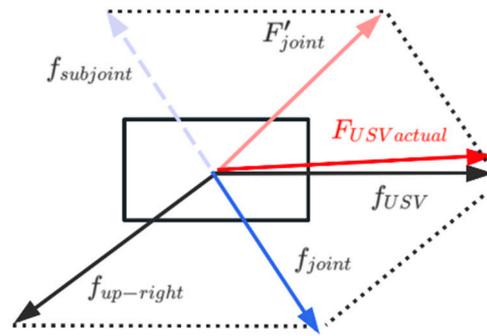


Figure 9. Illustrative diagram of actual resultant force of USV.

As can be seen from Figure 9, there is a certain angle error between the actual force and the actual direction of movement. To address this error, the grid map’s angle range is set. When the error angle is within the range of  $\phi \in (-\pi/4, \pi/4)$ , the USV selects the middle grid point within the error range as the target point; when the error angle is within the range of  $\phi \in (-\pi/3, -\pi/4) \cup (\pi/4, \pi/3)$ , there are multiple reachable grid points within this angle range, which may have a certain interference on the final actual target point position. Within this range, the error angle is associated with the grid point serial number, as specified in Formula (17).

$$\begin{cases} x = \text{mod}(c, n) - 0.5 \\ y = n + 0.5 - \text{ceil}(c/n) \pm (19.5 - x) \times \arctan\phi, \phi \in (-\frac{\pi}{4}, \frac{\pi}{4}) \\ x = \text{mod}(c, n) - 0.5 \\ y = [n + 0.5 - \text{ceil}(c/n)] \times f_{Ad}, \phi \in (-\frac{\pi}{3}, -\frac{\pi}{4}) \cup (\frac{\pi}{4}, \frac{\pi}{3}) \end{cases} \quad (17)$$

In this equation, the function  $f$  is `Advanced_sprcv`. When the error angle is  $\phi \in (-\pi/3, -\pi/4) \cup (\pi/4, \pi/3)$ , the number of grid errors is greater than or equal to 2. If the formula insists on using arctangent for projection, although the results can be obtained, there are multiple errors and obstacles in the projection process. Therefore, the function is used to apply the smooth optimization in advance, as shown in Figure 10.

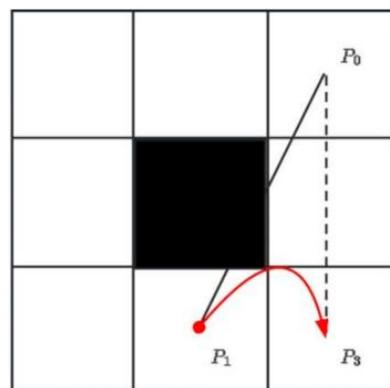


Figure 10. `Advanced_sprcv` optimization of large-angle error path.

When the error angle exceeds  $\pm \frac{\pi}{3}$ , this error cannot be eliminated, and it is regarded as a path exceeding the reachable range. Then, the algorithm returns to the previous node to re-optimize.

(4) Path-smoothing strategy

In this study, the originally planned path for unmanned surface vehicles (USVs) is depicted on a raster map as a sequence of disjointed line segments, which are interconnected sequentially at the raster’s center according to the coordinates of the planned path points.

This configuration results in sharp peak inflection points at the junctures of the path, posing a risk of damaging the USV’s hardware facilities. Moreover, these abrupt inflections hinder the USV’s ability to accurately follow the path, leading to unnecessary energy expenditure. To address this issue, this paper first proposes the utilization of quadratic Bezier curves to smooth out the sharp peaks, enhancing path continuity and reducing mechanical stress on the USVs. Recognizing the potential for collision, the study further incorporates the “spcrv” function, which ensures obstacle avoidance while maintaining the path’s smoothness. Bezier curves are mathematically represented as a series of connected control points, forming line segments that collectively define the smooth path, as illustrated in Figure 11. This approach not only mitigates the risk of hardware damage but also optimizes the energy efficiency of USVs by providing a more navigable path.

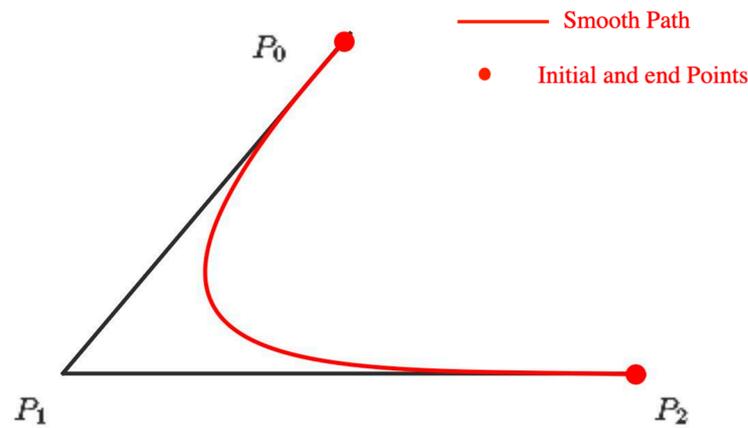


Figure 11. Quadratic Bezier curve.

The definition formula of an n-order Bezier curve (18) is

$$P_{[t_0,t_1]}(t) = \sum_{i=0}^n B_i^n(t)P_i, t \in [0,1] \tag{18}$$

where  $t$  is a location parameter,  $P_i$  represents control points, leading to  $P(t_0) = P_0$  and  $P(t_1) = P_n$ , and  $B_i^n(t)$  is the Bernstein polynomial given by the following formula:

$$B_i^m(t) = \binom{m}{i} t^i (1-t)^{m-i}, i = 0, 1, \dots, m, \tag{19}$$

$$P(t) = P_0(1-t) + P_1t \tag{20}$$

According to Equation (20), the linear Bezier curve represented by two points does not contain slope factors, so they form a segment in which the starting point and the endpoint are specified by the first and last points. The linear Bezier curve can be specified in the form of  $P(t) = (x(t), y(t))$ , where

$$x(t) = x_0(1-t) + x_1t \tag{21}$$

$$y(t) = y_0(1-t) + y_1t \tag{22}$$

$$P(t) = P_0(1-t)^2 + 2P_1t(1-t) + P_2t^2 \tag{23}$$

According to Equation (23), the quadratic Bezier curve represented by three points forms a parabolic figure, because they form a quadratic equation. The quadratic Bezier curve can be specified in the form of  $P(t) = (x(t), y(t))$ , where

$$x(t) = x_0(1-t)^2 + 2x_1t(1-t) + x_2t^2 \tag{24}$$

$$y(t) = y_0(1 - t)^2 + 2y_1t(1 - t) + y_2t^2 \tag{25}$$

$$L = \sqrt{(y_n - y_{n-1})^2 + (x_n - x_{n-1})^2} \tag{26}$$

According to Equation (26), the absolute distance between two nodes is given by  $L$ .

$$t = d/L \tag{27}$$

According to Equation (27), it is clear that the increment value is equal to the preset value  $d$  regardless of the distance between two nodes.

$$L_{total} = \sum_{m=1}^n L_m \tag{28}$$

Formula (28) is used to calculate the total length of the path from the starting point to the target point.

Since the path search algorithm generates a large number of path nodes and many turns, it cannot be used directly. Therefore, the optimal path was obtained using the Bezier smoothing process. The Bezier curve is used to smooth the sharp turning points in the path selected from redundant nodes. The smoothing steps are as follows:

1. Quadratic Bezier curves are applied to the nodes except for the starting point and ending point, similar to piecewise functions.
2. Whether obstacles are near the nodes is determined in the planning process, and thus, the curves are applied to these nodes safely.
3. As shown in Figure 12,  $P_w$  represents fixed control points, and  $P_{w+i}$  represents variable control points. When drawing the curve, Formulas (16)–(19) are used, and  $\| (P_w, P_{w+i}) \| + \| (P_w, P_{w-i}) \|$  in Formula (18) is set to  $L$ , which is Formula (29).

$$L = \| (P_w, P_{w+i}) \| + \| (P_w, P_{w-i}) \| \tag{29}$$

The curve drawn for small values up to  $i$ , corresponding to the obstacle, shows the safe curve.  $Obs$  represents obstacles in the environment, and  $P_w$  shows the nodes.  $P_{w+i}$  and  $P_{w-i}$  represent the waypoints between two nodes [34].

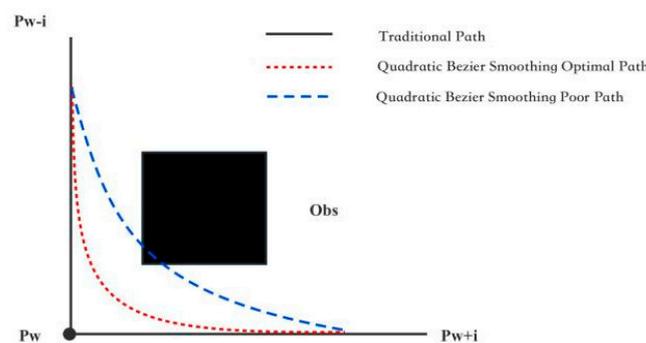


Figure 12. Quadratic Bezier smooth curve scenario.

The quadratic Bezier curve belongs to the convex optimization path-planning algorithm. In the smoothing process of the back-end optimization of path planning, quadratic optimization is the most common form, so it can ensure that there is an optimal solution under the current circumstances, and it can ensure that the local optimal solution is also the global optimal solution. However, in terms of obstacle avoidance, the reason why the situation in Figure 11 occurs is that this optimization can only ensure that there is no collision between waypoints and obstacles, but it cannot ensure that the polynomial connecting waypoints also has no collision with obstacles, and it cannot guarantee the ability to avoid dynamic obstacles. Therefore, in the case of complex environments and different forms of obstacles, the path can ensure smoothness but cannot guarantee planning safety.

To solve this problem, the convex optimization form needed to be changed, the number of control points on the way needed to be increased, and they needed to constitute high-order Bezier curves. However, high-order Bezier curves are prone to the Runge phenomenon, and there may be path redundancy in the process of finding target points. This improvement cannot eliminate the possibility of collision in the smoothing process, and it is crucial to use a dense sequence of functions with the sprcv function for smooth optimization to solve this problem. The function has a dense sequence  $f(tt)$  of points on the  $k$ -order uniform B-spline curve  $f$  with B-spline coefficients  $c$ , as shown in Formula (30).

$$f : t | \rightarrow \sum_{j=1}^n B(t - k/2 | j, \dots, j + k) c(j), \frac{k}{2} \leq t \leq n + \frac{k}{2} \tag{30}$$

In this formula,  $B(a, \dots, z)$  represents the B-spline with nodes  $a, \dots, z$ .  $n$  is the number of coefficients in  $c$ , i.e.,  $[d, n] = size(c)$ . The parameter interval filled with uniform sampling points  $tt$  is  $[k/2 \dots (n - k/2)]$ . The output is composed of the array  $f(tt)$ . After the smooth optimization is improved by sprcv based on Bezier smoothing, it ensures that the robot can successfully avoid obstacles and the path is smoother, as shown in Figure 12.

After Bezier optimization (Figure 13a), the path has achieved the smoothing goal, but there are still obvious problems in obstacle avoidance. After the sprcv improvement optimization, we have obtained a smooth and highly successful obstacle avoidance path (Figure 13b). In Figure 13, the red path is the shortest path found by the improved artificial bee colony algorithm in the 2D grid environment, the yellow path is the smooth path after the Bezier optimization curve, and the blue path is the smooth path after the improved sprcv curve optimization. By comparison, the path after Bezier curve optimization may encounter obstacles that cannot be avoided. Therefore, improved sprcv is added to improve the smoothness of the path and obtain an optimal path with high smoothness and safe obstacle avoidance.

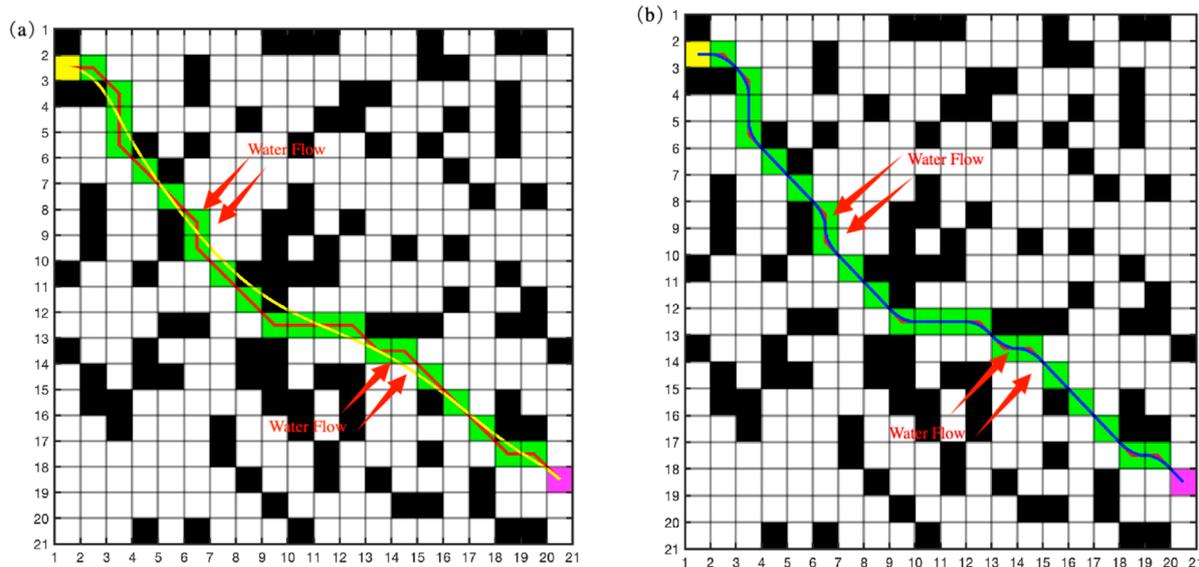


Figure 13. Comparative graph of (a) 2D Bezier optimization and (b) 2D SPRCV optimization.

Determining the obstacle avoidance ability of the USV requires testing not only whether it can successfully avoid static obstacles in the environment but also whether it can avoid static obstacles and completely avoid dynamic obstacles after adding dynamic obstacles to the same obstacle environment.

### 3.3. Experimental Facilities

The experimental equipment used in this article is mainly as follows: a Windows computer (Intel Core i5-10500H CPU, 64-bit operating system, 16GB memory, Hasee, Shenzhen, China) and a Mac computer (MacBook Pro, Apple M1 Pro, 10 cores, 16GB memory, Apple, Cupertino, CA, USA).

## 4. Simulation Experiment Analysis and Comparison

### 4.1. Dynamic Obstacle Avoidance Capability Testing of USV

Determining the obstacle avoidance ability of the USV requires testing not only whether it can successfully avoid static obstacles but also whether it can avoid both static and dynamic obstacles in the same obstacle environment.

In the traditional artificial bee colony algorithm, the path search efficiency is high, but it is only used for obstacle avoidance in a static obstacle environment, as shown in Figure 14. When the USV travels on the water surface, it may encounter both known obstacles and unknown moving obstacles. Therefore, it is necessary to improve the obstacle avoidance mechanism of the traditional artificial bee colony algorithm to enable it to normally avoid static obstacles and efficiently avoid dynamic obstacles on the water surface, as shown in Figure 15.

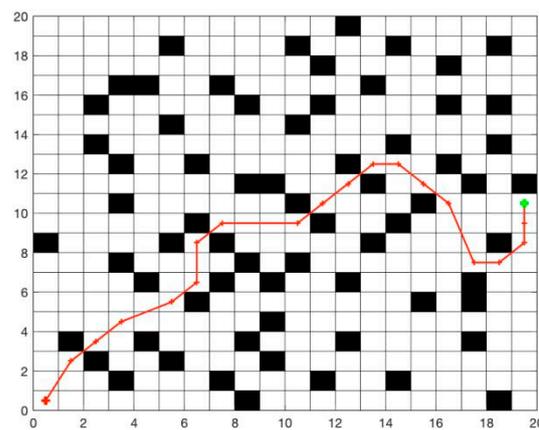


Figure 14. Path search graph of the two-dimensional artificial bee colony algorithm.

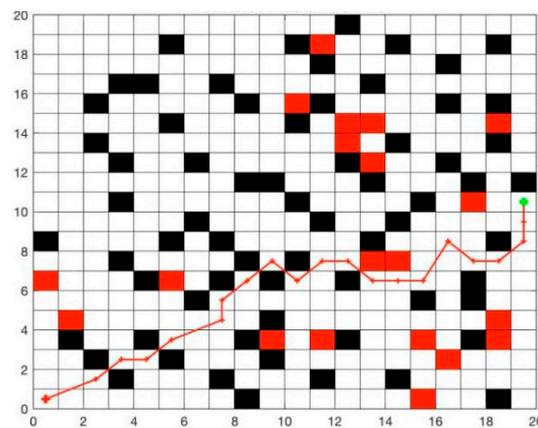
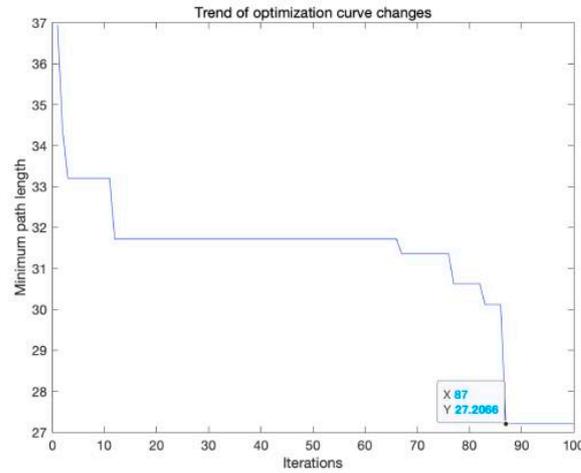


Figure 15. The improved bee colony algorithm search graph with the addition of dynamic obstacles.

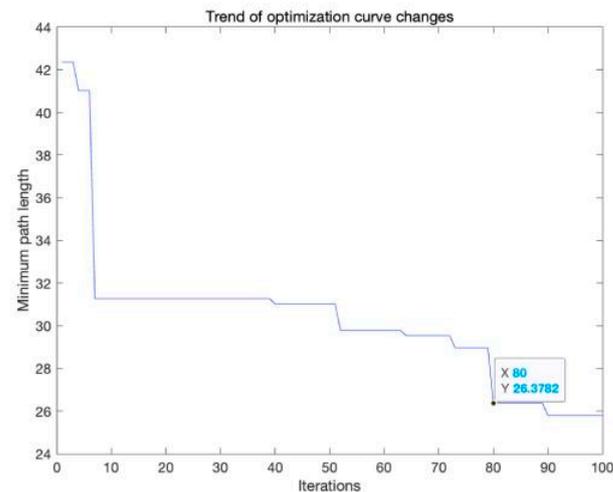
In the obstacle avoidance process of static obstacles, the updating of the population and the honey source will affect the search by honey bees to the target point, which leads to a fluctuation in the iteration times needed for the search for the optimal path in the traditional bee colony algorithm. As shown in Figure 16, under the premise of avoiding

static and dynamic obstacles, the traditional bee colony algorithm finds the shortest path after 87 iterations. After improving the algorithm, it can avoid obstacles in dynamic environments and find the shortest path after 80 iterations under the premise of avoiding both static and dynamic obstacles.



**Figure 16.** The iteration number of a population avoiding obstacles in the traditional bee swarm algorithm.

Based on the comparison of the number of iterations between traditional and improved artificial bee colony algorithms, it can be seen from Figures 16 and 17 that the improved bee colony reduces the excess foraging team on the optimization path and improves the optimization route of leading and following bees. Experimental verification shows that the improved algorithm improves the efficiency of searching for the shortest path under the requirement of successfully avoiding dynamic obstacles.



**Figure 17.** Improved bee swarm algorithm for obstacle avoidance: population iteration times.

#### 4.2. Comparison of Optimization Times for USV to Avoid Dynamic Obstacles in Different Environments

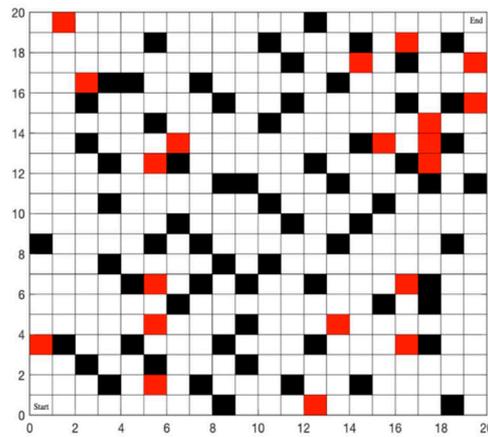
The USV can avoid static obstacles in time, but it also needs a certain reaction time to judge obstacles with the support of the algorithm. The traditional bee colony algorithm inevitably identifies and avoids obstacles in the process of path optimization, so the reaction time of the traditional algorithm is relatively long in a static environment, and there are many inflection points in the path of the map, which is also the reason for the longer optimization time of obstacle avoidance. As the USV cannot identify moving obstacles in

a new environment, the improved algorithm can achieve the ability to avoid obstacles in time. The results of the comparison of obstacle avoidance times on different grid maps are in Table 1.

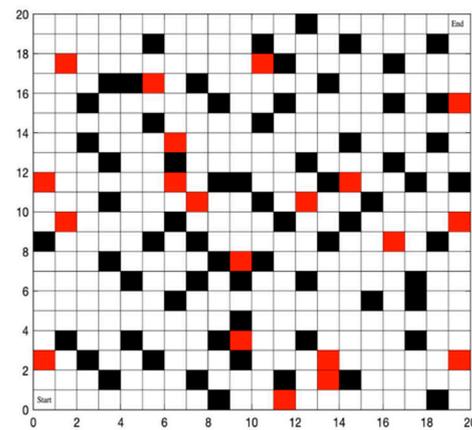
**Table 1.** The time consumed for path optimization in different dynamic environments using traditional and improved algorithms.

Compared Type of Algorithm	First Environmental Optimization Time/s	Second Environmental Optimization Time/s	Third Environmental Optimization Time/s	Fourth Environmental Optimization Time/s
ABC	1.10	1.34	0.85	1.03
DOASO-IABC	0.65	0.66	0.60	0.63

Figures 18–21 show the comparison of obstacle avoidance times for the two algorithms in different environments. Each grid map represents static obstacles in black and dynamic obstacles in red. The dynamic obstacles move randomly in four directions.



**Figure 18.** First grid map of environment for optimization path search.



**Figure 19.** Second grid map of environment for optimization path search.

In the grid environment with dynamic obstacles, the optimization time of the traditional algorithm is more than 1 s. After the improved algorithm, the optimization time of the USV is doubled. The experimental results verify that the improved algorithm has achieved the desired results. Therefore, the algorithm proposed in this paper has strong real-time obstacle avoidance ability.

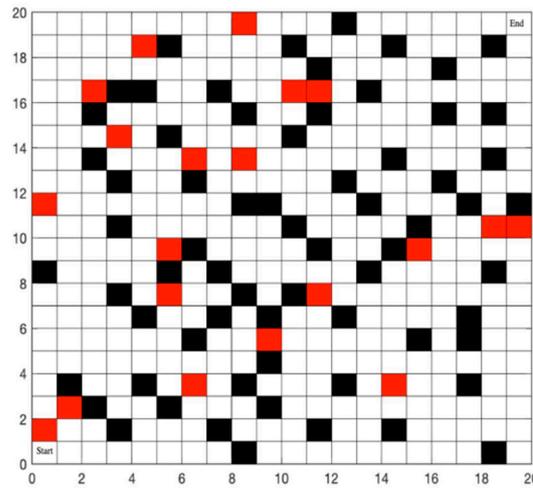


Figure 20. Third grid map of environment for optimization path search.

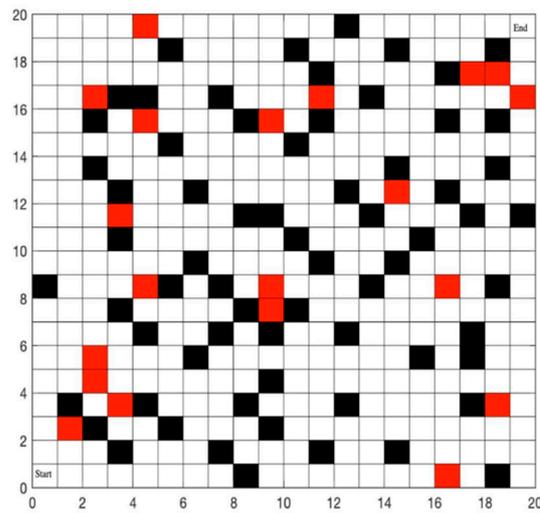


Figure 21. Fourth grid map of environment for optimization path search.

#### 4.3. Comparison of Optimal Path Length for Dynamic Obstacle Avoidance by USV in Different Environments

One of the factors that best reflect the advantages and disadvantages of the algorithm in path planning is the overall path length, which can be reflected through both the overall route length of the map and the intuitive path length value. The paths found by traditional algorithms are more tortuous and have more inflection points, so the overall path length is the longest. After improving the traditional algorithm and increasing the influence of actual water flow factors, although the path length shows a smoothing trend, due to the water flow, the angle deviation of USV travel is optimized by *Advanced\_sprcv*. Finally, after optimization, the path length obtained by the improved algorithm is shortened, and the effect is very good, as shown in Table 2 below.

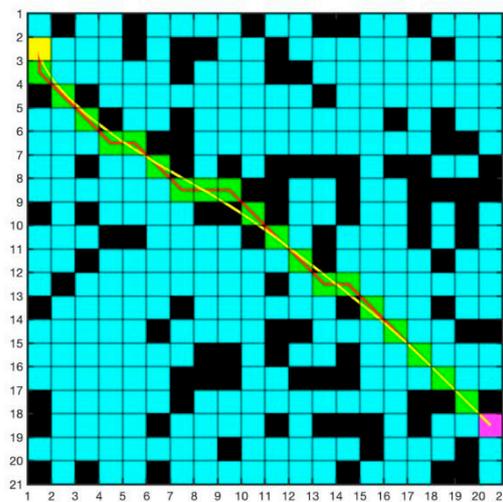
On the one hand, the path length is shortened by reducing the number of inflection points, which can turn the right angle at the turning point into a curve. On the other hand, the whole path is processed by a smoothing mechanism. The experimental results show that the improved algorithm has some success in optimizing the path length, and the optimization efficiency reaches 15%.

**Table 2.** Comparison of optimization path lengths of traditional bee colony algorithm and improved algorithm.

Compared Algorithm Type	First Environmental Path Length/Grid	Second Environmental Path Length/Grid	Third Environmental Path Length/Grid	Fourth Environmental Path Length/Grid
ABC	35.62	37.85	34.72	33.38
DOASO-IABC	33.04	33.55	30.79	29.73

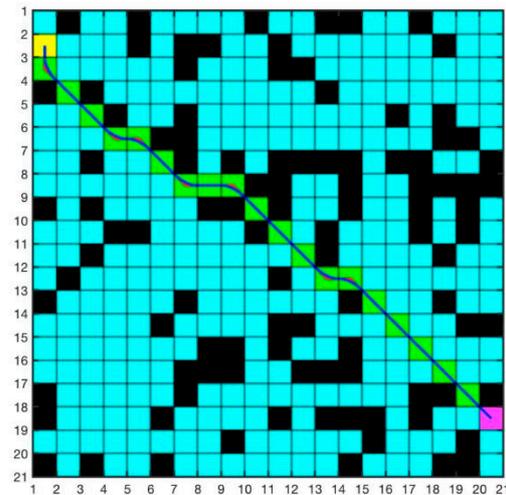
*4.4. Path-Smoothing Comparison*

The traditional bee colony algorithm for path planning deploys points, connects lines, searches for optimization by the bee colony, and determines the final route for each grid center position. Therefore, the overall path is mostly a broken line with many inflection points. Frequent changes in the direction of the USV in the water lead to the continuous adjustment of its power system, thereby increasing energy consumption, reducing the USV’s speed, prolonging the task execution time, and affecting the overall process efficiency. In response to this situation, a path-smoothing optimization mechanism is added to the traditional algorithm. After the experimental comparison, it is found that the overall smoothness of the path is improved after adding Improved\_sprcv, and obstacle avoidance can reach 100%. After the improved strategy is added, it is more practical to use Bezier curve smoothing paths than conventional ones. A comparison of the smoothness effect is shown in Figures 22 and 23.



**Figure 22.** The Bezier smooth optimization path diagram of artificial bee colony fusion.

In Figure 22, the red broken line is the shortest path found by the traditional artificial bee colony algorithm based on the starting point location. The yellow curve is the optimal path found by the algorithm after fusing Bezier curve optimization. By comparison, the improved path has completed the smoothing task, but it cannot avoid obstacles, especially moving obstacles. Therefore, this fusion does not meet practical requirements. Therefore, an improved artificial bee colony algorithm is proposed, which fuses the Improved\_sprcv smooth optimization mechanism, as shown in Figure 23. The red broken line is the optimal path of the traditional artificial bee colony algorithm, and the blue curve is the optimal path after smooth optimization. Compared with Bezier’s smooth optimization, this smooth mechanism can not only smooth the path but also avoid obstacles in time. According to experimental verification, this improvement strategy is suitable for practical application scenarios, and the improvement of the algorithm also improves the overall smoothness of the path.



**Figure 23.** Improved\_sprcv smooth optimization path map of bee swarm integration.

## 5. Conclusions

Effective path planning is crucial for unmanned surface vehicles (USVs) to navigate and avoid obstacles, especially amidst water currents and dynamic barriers. This study introduces a grid-based method for two-dimensional environmental modeling, focusing on overcoming these specific challenges. It highlights the importance of avoiding dynamic obstacles and achieving smooth path transitions. The paper presents a novel fusion swarm intelligence algorithm tailored for global path planning in known, dynamic aquatic environments. This algorithm integrates dynamic obstacle avoidance with optimization techniques, enhancing the ability to circumvent obstacles in fluid settings and identify locally optimal paths. This method significantly improves the accuracy and efficiency of path planning for USVs. When compared to traditional bee colony algorithms, the enhanced algorithm exhibits an 8% increase in the optimization path length, a 50% reduction in optimization time, and a nearly 100% obstacle avoidance capability in dynamic environments.

To further refine the path quality, we introduce an improved artificial bee colony algorithm (IABC). Through a series of simulation experiments, the superior performance of the proposed algorithm compared to traditional methods is demonstrated. It not only efficiently identifies the shortest optimal path but also ensures the path's high degree of smoothness and dynamic obstacle avoidance capabilities. It is important to note that the shortest path obtained may not necessarily represent the lowest energy consumption path, given the variable influence of water flow on energy consumption. The improvement of the algorithm in the article is not only applicable to USVs but also to underwater vehicles, unmanned aerial vehicles, and other facilities in the deep-sea field.

**Author Contributions:** S.C.: Methodology, Software, Writing—Review and Editing. L.F.: Formal Analysis, Validation, Writing. X.B.: Methodology and Investigation. Z.J.: Methodology. B.X.: Conceptualization, Data Curation. J.X.: Conceptualization, Supervision, Funding Acquisition, Project Administration. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported by the National Key R&D Program of China (No. 2022YFD2401100) and the Shanghai Science and Technology Committee Local Universities Capacity-Building Project (No. 22010502200). The authors would like to express their gratitude for the support of the Fishery Engineering and Equipment Innovation Team of Shanghai High-Level Local University.

**Institutional Review Board Statement:** No applicable.

**Informed Consent Statement:** No applicable.

**Data Availability Statement:** Data are contained within the article.

**Conflicts of Interest:** There are no conflicts to declare.

## References

1. Guo, X.; Ji, M.; Zhao, Z.; Wen, D.; Zhang, W. Global path planning and multi-objective path control for unmanned surface vehicle based on modified particle swarm optimization (PSO) algorithm. *Ocean Eng.* **2020**, *216*, 107693. [\[CrossRef\]](#)
2. Zhang, H.; Tao, Y.; Zhu, W. Global Path Planning of Unmanned Surface Vehicle Based on Improved A-Star Algorithm. *Sensors* **2023**, *23*, 6647. [\[CrossRef\]](#)
3. Li, C.; Huang, X.; Ding, J.; Song, K.; Lu, S. Global path planning based on a bidirectional alternating search A\* algorithm for mobile robots. *Comput. Ind. Eng.* **2022**, *168*, 108123. [\[CrossRef\]](#)
4. Li, H.; Zhao, T.; Dian, S. Forward search optimization and subgoal-based hybrid path planning to shorten and smooth global path for mobile robots. *Knowl.-Based Syst.* **2022**, *258*, 110034. [\[CrossRef\]](#)
5. Xia, G.; Han, Z.; Zhao, B.; Liu, C.; Wang, X. Global Path Planning for Unmanned Surface Vehicle Based on Improved Quantum Ant Colony Algorithm. *Math. Probl. Eng.* **2019**, *2019*, 2902170. [\[CrossRef\]](#)
6. Teli, T.A.; Wani, M.A. A fuzzy based local minima avoidance path planning in autonomous robots. *Int. J. Inf. Technol.* **2020**, *13*, 33–40. [\[CrossRef\]](#)
7. Lyridis, D.V. An improved ant colony optimization algorithm for unmanned surface vehicle local path planning with multi-modality constraints. *Ocean Eng.* **2021**, *241*, 109890. [\[CrossRef\]](#)
8. Wang, Z.; Li, G.; Ren, J. Dynamic path planning for unmanned surface vehicle in complex offshore areas based on hybrid algorithm. *Comput. Commun.* **2020**, *166*, 49–56. [\[CrossRef\]](#)
9. Zhang, L.; Zhang, Y.; Li, Y. Mobile Robot Path Planning Based on Improved Localized Particle Swarm Optimization. *IEEE Sens. J.* **2020**, *21*, 6962–6972. [\[CrossRef\]](#)
10. Chen, Z.; Zhang, Y.; Zhang, Y.; Nie, Y.; Tang, J.; Zhu, S. A Hybrid Path Planning Algorithm for Unmanned Surface Vehicles in Complex Environment with Dynamic Obstacles. *IEEE Access* **2019**, *7*, 126439–126449. [\[CrossRef\]](#)
11. Shi, K.; Wu, Z.; Jiang, B.; Karimi, H.R. Dynamic path planning of mobile robot based on improved simulated annealing algorithm. *J. Frankl. Inst.* **2023**, *360*, 4378–4398. [\[CrossRef\]](#)
12. Zhang, Z.; Qiao, B.; Zhao, W.; Chen, X. A Predictive Path Planning Algorithm for Mobile Robot in Dynamic Environments Based on Rapidly Exploring Random Tree. *Arab. J. Sci. Eng.* **2021**, *46*, 8223–8232. [\[CrossRef\]](#)
13. Han, S.; Wang, L.; Wang, Y.; He, H. A dynamically hybrid path planning for unmanned surface vehicles based on non-uniform Theta\* and improved dynamic windows approach. *Ocean Eng.* **2022**, *257*, 111655. [\[CrossRef\]](#)
14. Wang, C.; Yang, X.; Li, H. Improved Q-Learning Applied to Dynamic Obstacle Avoidance and Path Planning. *IEEE Access* **2022**, *10*, 92879–92888. [\[CrossRef\]](#)
15. Quan, Y.; Ouyang, H.; Zhang, C.; Li, S.; Gao, L.-Q. Mobile Robot Dynamic Path Planning Based on Self-Adaptive Harmony Search Algorithm and Morphin Algorithm. *IEEE Access* **2021**, *9*, 102758–102769. [\[CrossRef\]](#)
16. Chen, Y.; Bai, G.; Zhan, Y.; Hu, X.; Liu, J. Path Planning and Obstacle Avoiding of the USV Based on Improved ACO-APF Hybrid Algorithm with Adaptive Early-Warning. *IEEE Access* **2021**, *9*, 40728–40742. [\[CrossRef\]](#)
17. Wang, N.; Jin, X.; Er, M.J. A multilayer path planner for a USV under complex marine environments. *Ocean Eng.* **2019**, *184*, 1–10. [\[CrossRef\]](#)
18. Ma, Y.; Mao, Z.; Wang, T.; Qin, J.; Ding, W.; Meng, X. Obstacle avoidance path planning of unmanned submarine vehicle in ocean current environment based on improved firework-ant colony algorithm. *Comput. Electr. Eng.* **2020**, *87*, 106773. [\[CrossRef\]](#)
19. Guo, S.; Chen, M.; Pang, W. Path Planning for Autonomous Underwater Vehicles Based on an Improved Artificial Jellyfish Search Algorithm in Multi-Obstacle Ocean Current Environment. *IEEE Access* **2023**, *11*, 31010–31023. [\[CrossRef\]](#)
20. Zhou, Y.; Gong, Y.; Geng, X.; Li, D.; Gao, B.; Li, C. An Algorithm for Path Planning of Autonomous Ships Considering the Influence of Wind and Wave. *J. Phys. Conf. Ser.* **2021**, *2083*, 032028. [\[CrossRef\]](#)
21. Yang, J.; Huo, J.; Xi, M.; He, J.; Li, Z.; Song, H.H. A Time-Saving Path Planning Scheme for Autonomous Underwater Vehicles with Complex Underwater Conditions. *IEEE Internet Things J.* **2022**, *10*, 1001–1013. [\[CrossRef\]](#)
22. Wang, Z.; Liang, Y.; Gong, C.; Zhou, Y.; Zeng, C.; Zhu, S. Improved Dynamic Window Approach for Unmanned Surface Vehicles' Local Path Planning Considering the Impact of Environmental Factors. *Sensors* **2022**, *22*, 5181. [\[CrossRef\]](#)
23. Zhu, D.; Yang, S.X. Path planning method for unmanned underwater vehicles eliminating effect of currents based on artificial potential field. *J. Navig.* **2021**, *74*, 955–967. [\[CrossRef\]](#)
24. Yuan, X.; Tong, C.; He, G.; Wang, H. Unmanned Vessel Collision Avoidance Algorithm by Dynamic Window Approach Based on COLREGs Considering the Effects of the Wind and Wave. *J. Mar. Sci. Eng.* **2023**, *11*, 1831. [\[CrossRef\]](#)
25. Shen, Z.; Ding, W.; Liu, Y.; Yu, H. Path planning optimization for unmanned sailboat in complex marine environment. *Ocean Eng.* **2023**, *269*, 113475. [\[CrossRef\]](#)
26. Fu, J.; Lv, T.; Li, B. Underwater Submarine Path Planning Based on Artificial Potential Field Ant Colony Algorithm and Velocity Obstacle Method. *Sensors* **2022**, *22*, 3652. [\[CrossRef\]](#)
27. Dian, S.; Zhong, J.; Guo, B.; Liu, J.; Guo, R. A smooth path planning method for mobile robot using a BES-incorporated modified QPSO algorithm. *Expert Syst. Appl.* **2022**, *208*, 118256. [\[CrossRef\]](#)
28. Gu, Q.; Zhen, R.; Liu, J.; Li, C. An improved RRT algorithm based on prior AIS information and DP compression for ship path planning. *Ocean Eng.* **2023**, *279*, 114595. [\[CrossRef\]](#)
29. Wang, W.; Zhao, J.; Li, Z.; Huang, J. Smooth Path Planning of Mobile Robot Based on Improved Ant Colony Algorithm. *J. Robot.* **2021**, *2021*, 4109821. [\[CrossRef\]](#)

30. Xu, L.; Cao, M.; Song, B. A new approach to smooth path planning of mobile robot based on quartic Bezier transition curve and improved PSO algorithm. *Neurocomputing* **2021**, *473*, 98–106. [[CrossRef](#)]
31. Song, R.; Liu, Y.; Bucknall, R. Smoothed A\* algorithm for practical unmanned surface vehicle path planning. *Appl. Ocean Res.* **2018**, *83*, 9–20. [[CrossRef](#)]
32. Feng, K.; He, X.; Wang, M.; Chu, X.; Wang, D.; Yue, D. Path Optimization of Agricultural Robot Based on Immune Ant Colony: B-Spline Interpolation Algorithm. *Math. Probl. Eng.* **2022**, *2022*, 2585910. [[CrossRef](#)]
33. Zhai, L.; Zhang, X.; Zhang, X.; Wang, C. Local dynamic obstacle avoidance path planning algorithm for unmanned vehicles based on potential field method. *J. Beijing Inst. Technol.* **2022**, *42*, 696–705. [[CrossRef](#)]
34. Zhai, Z.; Dai, Y.; Zhou, P.; Liu, W. Path planning for mobile robots based on improved ant colony algorithm. *Comb. Mach. Tool Autom. Process. Technol.* **2023**, 5–9.

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.