Journal of
Marine Science and Engineering

MDPI

Article

Physical Consistent Path Planning for Unmanned Surface Vehicles under Complex Marine Environment

Fang Wang [1,2], Yong Bai [3] and Liang Zhao [3,*]

[1] School of Information and Electrical Engineering, Hangzhou City University, Hangzhou 310015, China; wangfang_hz@163.com
[2] Science and Technology on Underwater Vehicle Technology Laboratory, Harbin Engineering University, Harbin 150001, China
[3] College of Civil Engineering and Architecture, Zhejiang University, Hangzhou 310058, China; baiyong@zju.edu.cn
[*] Correspondence: liamzhao13@zju.edu.cn

Abstract: The increasing demand for safe and efficient maritime transportation has underscored the necessity of developing effective path-planning algorithms for Unmanned Surface Vehicles (USVs). However, the inherent complexities of the ocean environment and the non-holonomic properties of the physical system have posed significant challenges to designing feasible paths for USVs. To address these issues, a novel path planning framework is elaborately designed, which consists of an optimization model, a meta-heuristic solver, and a Clothoid-based path connector. First, by encapsulating the intricate nature of the ocean environment and ship dynamics, a multi-objective path planning problem is designed, providing a comprehensive and in-depth portrayal of the underlying mechanism. By integrating the principles of the candidate set random testing initialization and adaptive probability set, an enhanced genetic algorithm is devised to fully exploit the underlying optimization problem in constrained space, contributing to the global searching ability. Accounting for the non-holonomic constraints, the fast-discrete Clothoid curve is capable of maintaining and improving the continuity of the path curve, thereby promoting strong coordination between the planning and control modules. A thorough series of simulations and comparisons conducted in diverse ocean scenarios has conclusively demonstrated the effectiveness and superiority of the proposed path planning framework.

Keywords: path planning; unmanned surface vehicles; path smoothing; multi-objective; genetic algorithm

check for updates

Citation: Wang, F.; Bai, Y.; Zhao, L. Physical Consistent Path Planning for Unmanned Surface Vehicles under Complex Marine Environment. *J. Mar. Sci. Eng.* **2023**, *11*, 1164. https://doi.org/10.3390/jmse11061164

Academic Editor: Sergei Chernyi

Received: 11 May 2023
Revised: 23 May 2023
Accepted: 30 May 2023
Published: 1 June 2023

1. Introduction

With artificial intelligence at the helm, the advent of unmanned surface vehicles (USVs) has garnered significant attention, fueled by their potential to revolutionize maritime operations by enhancing safety and efficiency [1–6]. However, the successful deployment of USVs depends on the development of autonomous technology, which refers to the ability of these vehicles to plan and execute their missions in complex environments without human intervention, thereby enabling safe and efficient navigation [7,8]. Generally speaking, the most common approaches that contribute to the autonomous level of USVs are perception, localization and mapping, path planning and decision-making, and control system design. Central to achieving autonomy in USVs is the challenge of path planning, which involves determining an optimal path for the vehicle to traverse in order to accomplish its mission objectives while adhering to a set of predetermined rules and regulations. Compared to other types of autonomous vehicles, such as unmanned ground vehicles (UGVs), USV path planning may incorporate specialized techniques to handle challenges, such as wave prediction models, collision avoidance strategies for vessels, or algorithms, that account for hydrodynamic effects on the vehicle's motion. This task is particularly challenging due to the dynamic nature of maritime environments, which are subject to constantly changing weather conditions, currents, and other environmental factors that can impact navigation [9]. Achieving this goal requires the development of sophisticated path planning algorithms that enable these vehicles to navigate complex environments with minimal human intervention, paving the way for a future of safe and efficient maritime navigation [10].

The field of path planning for USVs has been an active area of research in recent years, with numerous studies investigating the development of effective planning strategies for USVs. In general, two primary categories of path planning algorithms have been proposed: global approaches and local approaches [7]. Global approaches involve the generation of a complete path for the USV based on prior knowledge of the environment, usually represented as a map. Such methods typically employ high-level planning techniques that treat the USV as a point object, neglecting its maneuverability and physical constraints. These methods are, therefore, more suitable for planning routes for long-distance voyages, where the emphasis is on efficient and safe navigation over extended periods. In contrast, local approaches generate a path by utilizing local information collected during the mission, enabling the USV to adapt to unexpected obstacles or changes in the environment. These methods fully consider the physical bounds of the USV's mechanical system, leading to more precise tracking performance for the low-level controller. Although the design of such methods is generally more complex, as it requires the integration of high-level planning and low-level control techniques to ensure effective operation, it is more applicable in practice.

Presently, there is a growing affinity for deterministic approaches in path planning, with various methods, such as A* and D* lite, basking in the limelight of scientific popularity. In particular, Yu and Wang [11] have put forward a hybrid algorithm that fuses artificial potential field (APF) and D* lite to navigate complex environments. This approach not only minimizes time cost but also enhances path safety through the APF. Nonetheless, it overlooks disturbances and energy consumption. Similarly, Yu et al. [12] have proposed an improved D* lite that reduces expanded nodes, validated via simulations. However, the simulation fails to consider ship dynamics, smoothness, and safety. Meanwhile, Song et al. [13] have utilized various smoothing techniques to mitigate the jagged effect in A*, which has been demonstrated to be effective through both experiments and simulations, making it a practical choice. Furthermore, Shah and Gupta [14] have presented a quadtree representation of the marine environment, which accelerates the A* algorithm without significantly sacrificing solution optimality, as shown in simulations. To facilitate path planning for working ships in offshore wind farms, Xie et al. [15] devised a multi-direction A* algorithm modified by an artificial potential field. Compared with the real-case trajectory, the minimum distance to the wind turbines has increased, and the path length outside the wind farm decreased dramatically. To solve the path planning problem under changing environments with multiple dynamic obstacles, Yao et al. [16] proposed an Improved D* lite algorithm, which has demonstrated its efficacy in real-time path planning through simulation results. Although deterministic approaches have emerged as popular and reliable methods for path planning, these methods can be computationally expensive, particularly when operating in large and complex environments. This can have a significant impact on their performance, making them less suitable for real-time applications.

As a result, the meta-heuristic algorithms, including ant colony optimization (ACO) [17], particle swarm optimization (PSO) [18], and genetic algorithms (GA) [19], have emerged as promising alternatives for path planning in marine robotics. These algorithms offer a set of high-level strategies to search for solutions, allowing them to optimize paths while considering multiple objectives with a comparatively low computational burden. Considering the effects of currents, Krell et al. [20] devised an improved PSO method implemented in visibility graphs. For the safe navigation of ships, a quasi-reflection-based PSO was proposed by Xue [21]. Incorporating the environmental loads, a hierarchical path planning framework based on GA is developed by Wang and Xu [10]. For rapid path generation, a leader-vertex ant colony optimization algorithm is proposed by Liang et al. [22], which ensures a leader of the ant colony and optimizes the route by vertex method. For both global and local path planning, a series of studies on artificial fish swarm algorithms have been conducted by Zhao et al. [7,8,23]. Under different current distribution, a comprehensive study has been conducted by Ma et al. [24] using multi-objective dynamic augmented PSO. Organically bridging the planning and tracking, Wang et al. [25] devised an elite-duplication GA (EGA) strategy to optimally generate sparse waypoints in a constrained space. However, meta-heuristic algorithms commonly encounter a significant hurdle with regard to their global searching capabilities, as they are prone to be trapped in local minima or suboptimum, thereby impeding the identification of the global optimum necessary for producing high-quality trajectories. Additionally, the computational efficiency of existing methods is not satisfactory enough to facilitate efficient path generation within high-dimensional configuration spaces [26]. Therefore, there is an imperative need for innovative techniques that can enhance the global searching capability and convergence rate of meta-heuristic algorithms for path planning.

In addition, a noteworthy limitation of most existing methods is the neglect of the non-holonomic constraints of the vehicle, which can lead to paths that are potentially infeasible. Specifically, the USV, being a non-holonomic robot, often functions as an underactuated system during its missions, resulting in limited maneuverability and motion flexibility [27,28]. Analogous to unmanned ground vehicles (UGVs), this restricts the USV's motion to forward velocity and manipulation of the heading angle to attain its desired position, thus precluding lateral movement. Consequently, it is of paramount importance to ensure smooth and continuous transitions of yaw and curvatures at turning points in order to devise an effective trajectory for a USV. For instance, sharp turns may be deemed unfeasible for a USV due to the significant sideslip that ensues, deviating from the planned path. Thus, the motion dynamics of the USV should be meticulously accounted for in the path planning process [10].

Inspired by the aforementioned literature review, this paper proposes a novel GA-variant meta-heuristic algorithm in combination with a fast-discrete Clothoid curve to optimize path generation. The main contributions of this paper are illustrated as follows:

- By capturing the non-holonomic nature of USVs and the intricate ocean dynamics, a sophisticated optimization model is carefully devised for the path planning problem, whereby the effects of currents, increments of curvatures, and constraints of physical system are addressed jointly.
- Introducing the random testing initialization algorithm and the adaptive design in the selection procedure, the proposed GA-variant facilitates strong global searching capabilities and a fast convergence rate, thereby contributing to the optimal generation of waypoint sequence.
- Accommodating the non-holonomic constraints, the fast-discrete Clothoid curve is able to preserve and enhance the continuity of the path curve, resulting in robust coordination between the planning and control module.

This paper is organized in the following structures: Section 2 presents the detailed modeling of the environment, USV, and the optimization model. Section 3 introduces the methodology. Illustrative simulation results are shown in Section 4. The conclusion is drawn in Section 5.

2. Problem Formulation

2.1. Environment Model

In this research, we consider the marine surface area represented as $\mathscr{M}$ in the Euclidean space $\mathbb{R}^2$. $\mathscr{M}$ is divided into obstacle area $\mathscr{M}_o$ and obstacle-free motion area $\mathscr{M}_f$, respectively. The relationship between these two grids is illustrated in Equation (1). The path P of the USV is defined as a sequence of connected elementary waypoints, denoted by $\boldsymbol{p}_i (i = 1, 2, 3, \ldots, m)$. By following the path P, the USV moves from the initial position $\boldsymbol{p}_S(x_S, y_S)$ to the final position $\boldsymbol{p}_E(x_E, y_E)$ while avoiding numerous obstacle areas $\mathscr{M}_o$ ($\mathscr{M}_o = \{O_1,\ O_2, \ldots, O_k\}$, where k denotes the number of obstacles).

$$\mathscr{M}_f + \mathscr{M}_o = \mathscr{M} \tag{1}$$

Accordingly, to guarantee the safety, the generated path should be restricted to $\mathscr{M}_f$ domain which is given as:

$$P = \cup_{i=1}^{m} \boldsymbol{p}_i \subset \mathscr{M}_f \tag{2}$$

2.2. Currents Model

According to previous research [20], ocean currents have a significant impact on the energy consumption of USVs. Therefore, when carrying out activities, USVs tend to choose a path that allows them to take advantage of the currents. Two types of currents exist: fixed and time-varying. Fixed currents are common in inland water voyages while time-varying currents are present in large-scale ocean environments. Assuming that the velocity of the USV in the body frame is $\boldsymbol{v} = [u,\ v,\ r]^T$ and the current velocity in the body frame is $\boldsymbol{v}_c$, the USV velocity, taking the effects of the currents into account, can be expressed as:

$$\boldsymbol{v}_r = \boldsymbol{v} + \boldsymbol{v}_c \tag{3}$$

2.3. USV Model

Typically, the motion of the USV can be regarded as a rigid body motion on the horizontal plane, as shown in Figure 1a, with three degrees of freedom: surge, sway, and yaw. Consequently, the state-space model for the USV, accounting for the impact of the current, can be derived as follows:

$$\begin{aligned} \dot{\boldsymbol{\eta}} &= \boldsymbol{R}(\psi)\boldsymbol{v_r} \\ \boldsymbol{M}\dot{\boldsymbol{v}}_{\boldsymbol{r}} + \boldsymbol{C}(\dot{\boldsymbol{v}}_{\boldsymbol{r}})\dot{\boldsymbol{v}}_{\boldsymbol{r}} + \boldsymbol{D}\boldsymbol{v_r} &= \boldsymbol{\tau} \end{aligned} \tag{4}$$

where the position and yaw angle in the earth-fixed inertial frame {n} are represented by the vector $\boldsymbol{\eta} = [x,\ y,\ \psi]^T$, while the relative velocities in the body-fixed frame {b} are denoted by $\boldsymbol{v_r} = [u_r,\ v_r,\ r]^T$, and the control signals are represented by the vector $\boldsymbol{\tau} = [\tau_u,\ 0,\ \tau_r]^T$. This paper considers the underactuated configuration of the USV, which means that the surge force and yaw moment are the only control forces. With these assumptions, the rotation matrix $\boldsymbol{R}(\psi)$, mass matrix $\boldsymbol{M}$, Coriolis matrix $\boldsymbol{C}$, and damping matrix $\boldsymbol{D}$ can be expressed as:

$$\begin{aligned} \boldsymbol{R}(\psi) &= \begin{bmatrix} cos(\psi) & -sin(\psi) & 0 \\ sin(\psi) & cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ \boldsymbol{M} &= \begin{bmatrix} m - X_{\dot{u}} & 0 & 0 \\ 0 & m - Y_{\dot{v}} & 0 \\ 0 & 0 & I_z - N_{\dot{r}} \end{bmatrix} = \begin{bmatrix} m_{11} & 0 & 0 \\ 0 & m_{22} & 0 \\ 0 & 0 & m_{33} \end{bmatrix} \\ \boldsymbol{C} &= \begin{bmatrix} 0 & 0 & -(m - Y_{\dot{v}})v \\ 0 & 0 & (m - X_{\dot{u}})u \\ (m - Y_{\dot{v}})v & -(m - X_{\dot{u}})u & 0 \end{bmatrix} \\ &= \begin{bmatrix} 0 & 0 & -m_{22}v \\ 0 & 0 & m_{11}u \\ m_{22}v & -m_{11}u & 0 \end{bmatrix} \\ \boldsymbol{D} &= \begin{bmatrix} d_{11} & 0 & 0 \\ 0 & d_{22} & 0 \\ 0 & 0 & d_{33} \end{bmatrix} \end{aligned} \tag{5}$$

To represent the USV's orientation relative to the earth-fixed inertial frame, we use a rotation matrix $\boldsymbol{R}(\psi)$ that transforms the body-fixed frame. The mass matrix $\boldsymbol{M} = \boldsymbol{M^T} > 0$ takes into account the USV's inertial properties and hydrodynamic added mass. The matrix $\boldsymbol{D}$ incorporates damping coefficients, while the Coriolis matrix $\boldsymbol{C}$ captures the Coriolis and centripetal effects and can be obtained from $\boldsymbol{M}$.

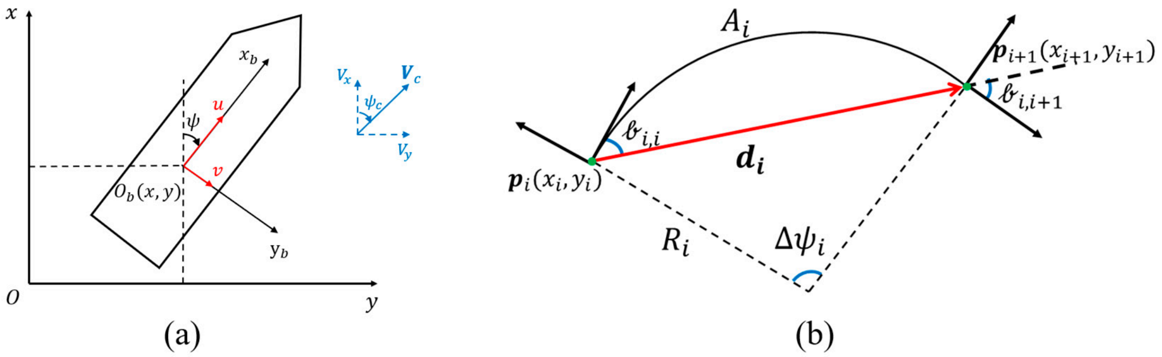

Figure 1. (**a**) Coordinate system; (**b**) Definition of a path curve.

Due to its non-holonomic nature, a USV typically has limited maneuverability and motion flexibility during most operations [28]. Similar to UGVs, the non-holonomic constraint restricts its lateral motion, meaning that the USV can only use its forward velocity while adjusting the heading angle to reach a desired position.

In Figure 1b, let $\boldsymbol{d}_i = [x_{i+1} - x_i,\ y_{i+1} - y_i, 0]^T$ denote the position vector between two points, and $\boldsymbol{p}_i$ and $\boldsymbol{p}_{i+1}$ represent the waypoints. The angles between $\boldsymbol{d}_i$ and $\boldsymbol{p}_i$ and $\boldsymbol{p}_{i+1}$ are defined as $\vartheta_{i,\ i}$ and $\vartheta_{i,\ i+1}$, respectively. In order to ensure a continuous path, the straight line and turning motions require that two consecutive positions $\boldsymbol{p}_i$ and $\boldsymbol{p}_{i+1}$ lie on a common arc of constant curvature, which can be expressed as:

$$\vartheta_{i,\ i} = \vartheta_{i,\ i+1} \tag{6}$$

Moreover, the turning angle at any point on the path must be restricted in the dynamic bounds, which gives:

$$\Delta\psi_i \leq \Delta\psi_{max} \tag{7}$$

2.4. Optimization Terms

2.4.1. Cruising Time

Since the path length and energy consumption objectives are interdependent, modifying the design variables affects both objectives equally. To jointly represent these objectives, we have adopted the cruising time t as the first objective. Let $\boldsymbol{v}_i$ be the velocity of the USV at $\boldsymbol{p}_i$ and $\boldsymbol{v}_c$ be the current velocity. The resultant velocity is denoted by $\boldsymbol{v}_r = \boldsymbol{v}_i + \boldsymbol{v}_c$. The cruising time t_i between $\boldsymbol{p}_i$ and $\boldsymbol{p}_{i+1}$ can be calculated as follows:

$$t_i = \frac{||\boldsymbol{p}_{i+1} - \boldsymbol{p}_i||}{\boldsymbol{v}_{i,c}} \tag{8}$$

Then, the total cruising time is calculated by:

$$\mathcal{T} = \sum_{i=1}^{m-1} t_i \tag{9}$$

The calculation of $\boldsymbol{v}_c$ is performed using the current distribution function. In most cases, $\boldsymbol{v}_i$ is considered to be a constant value in the same direction as $\boldsymbol{p}_i$. Therefore, t_i represents the nominal cost of travel time and does not accurately reflect the actual travel time.

2.4.2. Smoothness and Continuity

The additional cost incurred due to yaw is closely linked to the motion control performance of the USV. In order to enhance the smoothness of the trajectory, an objective function is introduced. To achieve this, constraint (9) is added as a quadratic penalty term to the objective function. The turning angle $\Delta\psi_i$ between waypoints $\boldsymbol{p}_i$ and $\boldsymbol{p}_{i-1}$ in the path P is calculated as $\Delta\psi_i = |\psi_i - \psi_{i-1}|$, where $\psi_i = atan((y_i - y_{i-1})/(x_i - x_{i-1}))$ and $\psi_{i-1} = atan((y_{i-1} - y_{i-2})/(x_{i-1} - x_{i-2}))$. The smoothest path is achieved when the changes in $\Delta\psi_i$ $(i = 2, 3, \ldots, m)$ and curvature $|\mathscr{C}_{i-1,\, i-1} - \mathscr{C}_{i-1,\, i}|$ are minimized. Hence, the objective function for achieving the smoothest path is defined as:

$$\vartheta = \sum_{i=2}^{m} \Delta\psi_i + \sum_{i=2}^{m} |\mathscr{C}_{i-1,\, i-1} - \mathscr{C}_{i-1,\, i}| \tag{10}$$

2.4.3. Path Safety

To ensure the safe movement of the USV, it is crucial to find a collision-free path that also maintains a safe distance from the obstacles. In addition to satisfying the conditions outlined in Equation (2), the minimum clearance from obstacles, denoted as d_i, is used to determine the safety of the solution. Specifically, we define two circular areas with radii d_{min} and d_{max} around each path waypoint $\boldsymbol{p}_i$. The distance between each path waypoint $\boldsymbol{p}_i$ and its closest obstacle O_i $(O_i \subset \mathscr{M}_o)$ is represented by $d_i = ||\boldsymbol{p}_i,\, O_i||, (i = 1, 2, 3, \ldots, m)$. The safety of each point along the path can then be evaluated by:

$$\mathscr{D}_i = \begin{cases} 0, & d_i \geq d_{max} \\ \frac{d_{max} - d_i}{d_{max} - d_{min}}, & d_{min} < d_i < d_{max} \\ 1, & d_i \leq d_{min} \end{cases},\ i = 1, 2, 3, \ldots, m \tag{11}$$

Therefore, to ensure the safety of the path, the third objective is to minimize the minimum value of $\mathscr{D}_i$, see the following:

$$\mathscr{D} = \text{argmin}\ \{\mathscr{D}_1,\ \mathscr{D}_2,\ \ldots,\ \mathscr{D}_i\},\ \ i = 1, 2, 3, \ldots, m \tag{12}$$

2.5. Problem Statement

The path planning model for the problem is formed as:

$$\min \mathscr{J} = \mathscr{T} + \vartheta + \mathscr{D} \tag{13}$$

s.t.

$$\begin{array}{l}
\mathscr{M}_f = \mathscr{M} - \mathscr{M}_o \\
P = \cup_{i=1}^{m} \boldsymbol{p}_i \subset \mathscr{M}_f,\ \ i = 1, 2, 3, \ldots, m \\
\boldsymbol{p}_1 = (x_1,\ y_1) = (x_S, y_S) \\
\boldsymbol{p}_m = (x_m, y_m) = (x_E, y_E) \\
\boldsymbol{v} + \nu_c \geq 0 \\
\boldsymbol{v}_r = \boldsymbol{v} + \boldsymbol{v}_c,\ \ i = 1, 2, 3, \ldots, m \\
\mathscr{C}_{i,\, i} = \mathscr{C}_{i,\, i+1},\ \ i = 1, 2, 3, \ldots, m-1 \\
\mathscr{t}_i = \frac{||\boldsymbol{p}_{i+1} - \boldsymbol{p}_i||}{\boldsymbol{v}_{i,c}},\ \ i = 1, 2, 3, \ldots, m-1 \\
\mathscr{T} = \sum\limits_{i=1}^{m-1} \mathscr{t}_i,\ \ i = 1, 2, 3, \ldots, m \\
\psi_i = atan((y_i - y_{i-1})/(x_i - x_{i-1})),\ \ i = 2, 3, \ldots, m \\
\Delta\psi_i = |\psi_i - \psi_{i-1}|,\ \ i = 2, 3, \ldots, m \\
\Delta\psi_i \leq \Delta\psi_{max} \\
\vartheta = \sum\limits_{i=2}^{m} \Delta\psi_i + \sum\limits_{i=2}^{m} |\mathscr{C}_{i-1,\, i-1} - \mathscr{C}_{i-1,\, i}|,\ \ i = 2, 3, \ldots, m \\
d_i = ||\boldsymbol{p}_i,\ O_i||, (i = 1, 2, 3, \ldots, m) \\
\mathscr{D}_i = \begin{cases} 0, & d_i \geq d_{max} \\ \frac{d_{max} - d_i}{d_{max} - d_{min}}, & d_{min} < d_i < d_{max} \\ 1, & d_i \leq d_{min} \end{cases},\ i = 1, 2, 3, \ldots, m \\
\mathscr{D} = argmin\ \{\mathscr{D}_1,\ \mathscr{D}_2,\ \ldots,\ \mathscr{D}_i\},\ \ i = 1, 2, 3, \ldots, m
\end{array} \tag{14}$$

3. Solver Design

3.1. Adaptive-Elite Genetic Algorithm

The genetic algorithm (GA) was initially proposed by Professor J. Holland in 1973 as a meta-heuristic optimization method. By simulating the evolutionary process of an artificial population, the GA manipulates each individual in the population through genetic operations, such as selection, crossover, and mutation. The process generates a new population with the best-performing individuals from the previous generation as the parents. The population evolves through several generations, and the individuals with the best fitness values are selected as the optimal solutions.

The GA's strength lies in its ability to search a large solution space using stochastic searches and evolutionary operations, such as crossover and mutation, making it effective in handling non-linear and non-convex optimization problems. Moreover, the GA's population size enables it to mitigate the impact of hyperparameter selection by allowing the algorithm to sample from a diverse set of solutions. Given that the optimization problem presented in this paper is an NP-hard nonlinear problem, we choose the GA as the primary framework.

3.1.1. Chromosome Representation

In evolutionary algorithms, chromosomes can be represented in various ways, such as binary-coded, real-coded, and decimal-coded. In our paper, we utilized the real-coded chromosome to directly represent the USV's path. Specifically, we use a sequence of points that begins at an origin position and ends at a destination point. Each point, denoted as $\boldsymbol{p}_i = (x_i, y_i)$, is saved along with its x and y coordinates and a pointer to the next point in the path. Figure 2 illustrates this representation.

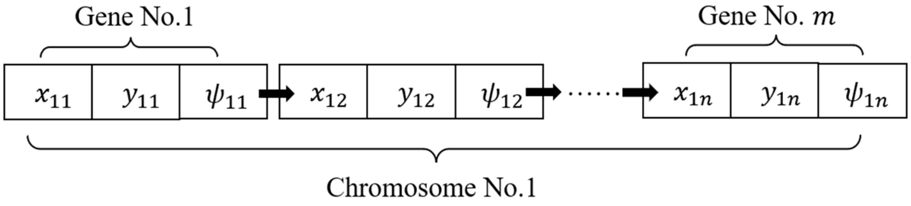

Figure 2. Definition of chromosome.

3.1.2. Initialization

The original population for the algorithm is obtained through population initialization. In the case of the original GA, a certain number of solutions are randomly generated in the solution space without the use of a heuristic function. This can lead to a random and unfocused solving process, resulting in a high proportion of poor solutions and low-quality genes in the population. This, in turn, requires a long convergence time during subsequent evolution and makes the solving process prone to being trapped in a local optimum.

To address this issue, we have developed a modified initialization method for GA inspired by failure analysis techniques used in software systems. Specifically, we have incorporated a candidate set adaptive random testing (ART) approach to improve the diversity of the initial population. By enhancing the initial diversity, the ART-based initialization method allows the GA to explore a broader range of potential solutions. This exploration can improve the algorithm's ability to escape local optima and discover better solutions in the search space. Consequently, it enhances the chances of finding high-quality solutions and can potentially accelerate the convergence toward optimal or near-optimal solutions. In summary, compared to standard random initialization, the ART-based initialization method in the GA offers the advantage of generating an initial population that is more diverse and better distributed throughout the search space. This increased diversity can facilitate improved exploration of the solution space and potentially lead to better overall performance and convergence in the GA.

The main steps of the initialization process are illustrated as follows:

Step 1: m candidate individuals $C = c_1, c_2, \ldots, c_m$ are randomly generated.

Step 2: The objective distances between each candidate individual and the current individuals in the population set $P = \boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_n$ are calculated.

Step 3: The shortest distance between each candidate individual and the population set is identified.

Step 4: The candidate individual with the maximum distance value is selected and added to the population set P.

3.1.3. Selection Operator

In a genetic algorithm, the selection operator is responsible for choosing individuals from a population for the crossover operator. This selection process is carried out based on a predefined regulation called the Roulette Wheel Selection (RWS) method. To perform Roulette Wheel Selection (RWS) in a genetic algorithm (GA), first, we compute the fitness values for each individual in the population and normalize them to obtain probabilities. Then, calculate the cumulative probabilities by summing up the normalized fitness values. After that, generate a random number between 0 and 1, and select individuals whose cumulative probability exceeds this random number. Repeat the selection process as needed to obtain the desired number of parents. Use the selected parents for crossover or recombination to generate offspring for the next generation. This iterative process allows individuals with higher fitness to have a greater chance of being selected, promoting the propagation of favorable traits in the GA. The RWS method ensures that the fitter individuals have a higher chance of being selected for the crossover, thus improving the overall quality of the population in the subsequent generation. The selection probability of each individual can be expressed as follows:

$$P^i_{select} = F(x_i) / \sum_{i=1}^{n} F(x_i) \tag{15}$$

where x_i denotes the individual and $f(x_i)$ is the corresponding fitness value. As can be seen from Equation (15), better individuals have more chances to be selected by RWS, which leads to better solutions.

3.1.4. Hybrid Crossover

Crossover operators are utilized to combine two solutions and generate a new offspring with better performance in terms of a predefined objective. These operators can be applied to solutions with the same or different number of waypoints. The first crossover operator involves calculating the mean of the two parent solutions to produce a new offspring.

$$x^f = \frac{x_1^{\boldsymbol{p}} + x_2^{\boldsymbol{p}}}{2}, \quad y^f = \frac{y_1^{\boldsymbol{p}} + y_2^{\boldsymbol{p}}}{2} \tag{16}$$

where two parents have gene coordinates denoted by $x_1^{\boldsymbol{p}}$, $y_1^{\boldsymbol{p}}$, and $x_2^{\boldsymbol{p}}$, $y_2^{\boldsymbol{p}}$, respectively. The gene coordinates of the offspring are represented by x^f and y^f. First, two parent chromosomes are selected according to the selection operator. Second, we select one of the parents as a reference chromosome. In this procedure, if the number of waypoints in the parents are the same, we choose the reference randomly. Otherwise, the one with smaller waypoint number is chosen. Then, waypoints of the offspring $\left(x^f, y^f\right)$ are calculated by taking the mean of each waypoint of the reference chromosome and the nearest waypoint of the other parent. To determine the gene coordinates of the offspring $\left(x^f, y^f\right)$, the average of each gene in the selected parent and the closest gene in the other parent are computed.

To enhance the variability of the population and explore the entire available space, the second crossover operator is utilized in which the two parents are randomly merged:

$$x^f = kx_1^{\boldsymbol{p}} + (1-k)x_2^{\boldsymbol{p}}, \quad y^f = ky_1^{\boldsymbol{p}} + (1-k)y_2^{\boldsymbol{p}} \tag{17}$$

where vector k consists of random numbers ranging from -1 to 1. When the number of genes differs between the parents, a similar approach to the first operator is used to combine genes with minimal distance. The primary aim of the first operator is to escape local optima, while the second one explores the environment randomly, preventing premature convergence.

3.1.5. Mutation Operator

The mutation operator in a Genetic Algorithm has a critical function in maintaining diversity within the population. The primary goal of the mutation operator is to randomly modify the value(s) of one or more genes within an individual's chromosome. By introducing such changes, the mutation operator assists in preventing the GA from becoming trapped in a local optimum, which would hinder the search for the global optimum. In the absence of the mutation operator, the GA may converge to a suboptimal solution that is in close proximity to the initial population. Hence, mutation serves as a crucial component of GA by promoting exploration of the search space and preventing premature convergence to suboptimal solutions. This paper introduces two mutation operators to facilitate the genetic process:

The first operator is a random mutation that selects one position on chromosomes and changes the value in the free space as shown in the following figure:

A different mutation operator is utilized to enhance the path's smoothness and length by adjusting the position of a gene. The operator integrates the present position ($\boldsymbol{p}_i$) of a gene with the directions towards the genes located on either side, $\boldsymbol{p}_{i-1}$ and $\boldsymbol{p}_{i+1}$, using the subsequent expressions:

$$\begin{aligned} x_i^f &= x_i^p + m\left(x_{i-1}^p - x_i^p\right) + n\left(x_{i+1}^p - x_i^p\right) \\ y_i^f &= y_i^p + m\left(y_{i-1}^p - y_i^p\right) + n\left(y_{i+1}^p - y_i^p\right) \end{aligned} \tag{18}$$

where m and n are random positive coefficients from 0 to 1. As illustrated in Figure 3b, this mutation operator leads to the creation of paths with shorter lengths and better smoothness. Combining both mutation operators result in a powerful tool that enhances both the searching and convergence capabilities of the algorithm.

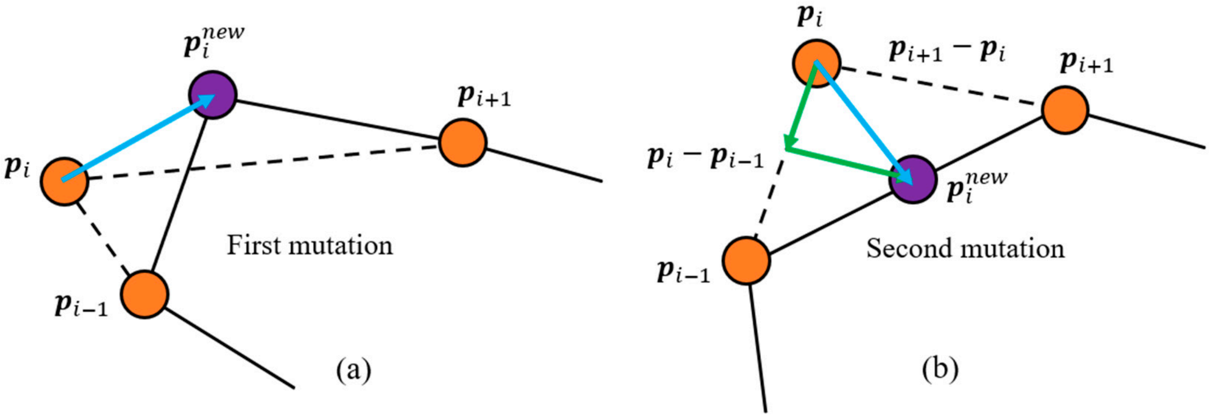

Figure 3. Mutation operators. (**a**) first mutation, (**b**) second mutation.

3.1.6. Fitness Design

In this paper, a multi-objective fitness function is devised. For this purpose, a weighted linear combination of the mentioned objectives is considered:

$$F = c_1 w_t \mathscr{T} + c_2 w_\theta \vartheta + c_3 w_d \mathscr{D} \tag{19}$$

The formula involves several parameters, including $\mathscr{T}$ (cruising time), ϑ (smoothness objective), and $\mathscr{D}$ (safety level), where w_t, w_θ, and w_d represent weight values, and their sum is equal to 1. To maintain consistency in the indicators' magnitudes, coefficients c_1, c_2, and c_3 have been set to 0.1, 1, and 100, respectively, as shown in the equation.

Selecting appropriate weight values is a vital aspect of the algorithm's performance. However, relying solely on empirical methods can be subjective. In an effort to achieve more balanced results, the Delphi weighting method [19] was employed to determine the weight of each indicator. As a result, the weight coefficients for cruising time, smoothness, and safety are 0.395, 0.275, and 0.330, respectively.

3.1.7. Determination of $\mathrm{p_c}$ and $\mathrm{p_m}$

The conventional Genetic Algorithm for USV path planning relies on two essential parameters, namely, the crossover rate and mutation rate, to regulate $\boldsymbol{p}_c$ and $\boldsymbol{p}_m$ of individuals in each iteration. However, using fixed values for these parameters may pose certain challenges. For instance, employing a large crossover and mutation probability can make it difficult to retain the best individuals, slow down population convergence, and consequently, delay the generation of the inspection path, thereby impacting operational efficiency. Conversely, a small $\boldsymbol{p}_c$ and $\boldsymbol{p}_m$ can negatively affect the searching process, leading to the local optimum. This, in turn, causes the USV to travel longer distances, reducing its efficiency.

To tackle the aforementioned challenges, a modified approach is suggested. This approach involves adjusting $\boldsymbol{p}_c$ and $\boldsymbol{p}_m$ during the algorithm execution. Specifically, in the early stages of the algorithm, $\boldsymbol{p}_c$ and $\boldsymbol{p}_m$ are increased to improve the global search ability, while in the final stage, the probabilities are decreased to facilitate good convergence. Adaptive probabilities allow the GA to dynamically adjust the rates of crossover and mutation based on the progress of the algorithm. Initially, higher probabilities promote exploration by encouraging diverse offspring. As the GA progresses, the probabilities can be reduced, shifting the focus towards exploitation of promising solutions. This balance between exploration and exploitation helps the GA efficiently search for optimal or near-optimal solutions.

To achieve this, adaptive functions are formulated as follows:

$$\boldsymbol{p}_c = \boldsymbol{p}_{c0} e^{-\frac{a}{F}} \tag{20}$$

where $\boldsymbol{p}_{c0}$ represents the initial $\boldsymbol{p}_c$, a is the scaling coefficient, and F here is the average fitness of the population. Similarly, the mutation probability can be obtained with same structure as follows:

$$\boldsymbol{p}_m = \boldsymbol{p}_{m0} e^{-\frac{b}{F}} \tag{21}$$

where $\boldsymbol{p}_{m0}$ represents the initial $\boldsymbol{p}_m$ and b is the scaling coefficient. These functions dynamically adjust the crossover probability based on the mean fitness degree of the population at each generation. As a result, the USV can achieve a balance between exploration and exploitation, leading to faster convergence and better results.

3.2. Fast-Discrete Clothoid Curve

To ensure real-time performance and accommodate the USV's kinematic constraints, we introduce a Fast-Discrete Clothoid Path (FDCP) to construct and connect the path. The FDCP employs a sequence of control points, referred to as waypoints, which are linked together using Clothoid segments. However, accurately generating Clothoids can be difficult due to their non-linear nature and multiple solutions. Thus, instead of directly computing the parameters of the Clothoid segments, our algorithm utilizes a variational approach that produces a polyline with linear discrete curvature, which approximates the Clothoid segment. This approach allows for efficient and precise path planning for the USV, while taking into account the vehicle's non-holonomic features.

To determine the position of intersection points, the following conditions must be met when inserting or updating point C between neighboring points B and D, as shown in Figure 4. To simplify the calculations, a normalized configuration is used where point B is located at $(-1, 0)$ and point D at $(1, 0)$. For each of the five control points denoted as

P, its left and right neighbors are identified as P_l and P_r, respectively. The insertion point C must satisfy the following conditions to approximate the Clothoids accurately between these control points:

- C must lie on the perpendicular bisector between B and D.
- The curvature at each point should vary linearly.

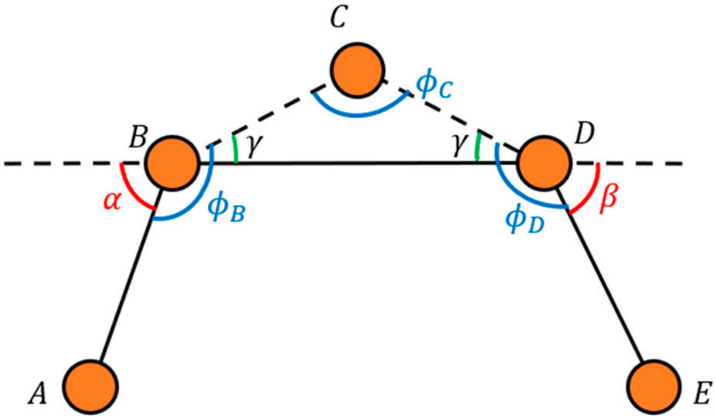

Figure 4. Locating point C.

From the abovementioned conditions, we have the following equation:

$$\rho_C = \frac{1}{2}(\rho_B + \rho_D) \tag{22}$$

where ρ denote the curvature at each point, it can be approximated by:

$$\rho_P = 2\frac{\pi - \phi_P}{|P_l P| + |PP_l|} \tag{23}$$

where ϕ_P is the angle between $|P_l P|$ and $|PP_l|$, and $|P_l P|$ is the scalar value of the length between P_l and P. According to the geometric relations in Figure 4, we obtain:

$$\begin{aligned} \phi_B &= \pi - \alpha + \gamma \\ \phi_C &= \pi - 2\gamma \\ \phi_D &= \pi - \beta + \gamma \end{aligned} \tag{24}$$

Substituting ϕ_P in previous equation using Equation (16), we have:

$$\begin{aligned} \rho_B &= 2\tfrac{\alpha - \gamma}{|AB| + |BC|} \\ \rho_C &= 2\tfrac{2\gamma}{|BC| + |CD|} \\ \rho_D &= 2\tfrac{\beta - \gamma}{|CD| + |DE|} \end{aligned} \tag{25}$$

As more and more points are inserted, the polyline gets refined and the angles between segments approach π. Therefore, with a large number of sample points, we can approximate $|BC| = |CD| = 1$. Solving Equation (14), γ can be obtained by:

$$\gamma = \frac{\beta(|AB| + 1) + \alpha(|DE| + 1)}{2|AB||DE| + 3(|AB| + |DE|) + 4} \tag{26}$$

Point C is now inserted on the perpendicular bisector between B and D in distance $|CD| \tan \gamma$. By iteratively inserting the intersection points (such as point C), we can approximate the Clothoid path with satisfactory computational performance.

Clothoid curves provide a continuous change in curvature, resulting in a smooth transition between straight segments and curved segments of a path. This helps reduce abrupt changes in the path and improves the vehicle's stability and comfort. Moreover, by gradually changing the curvature, Clothoid curves minimize lateral acceleration during turns. This reduces the forces acting on the USV, enhancing safety and stability during maneuvering. Additionally, Clothoid curves enable more precise and controlled maneuvering. They allow for gradual changes in heading angle, facilitating smooth turns, and transitions between different paths or waypoints.

The flowchart of the methodology is illustrated by Figure 5.

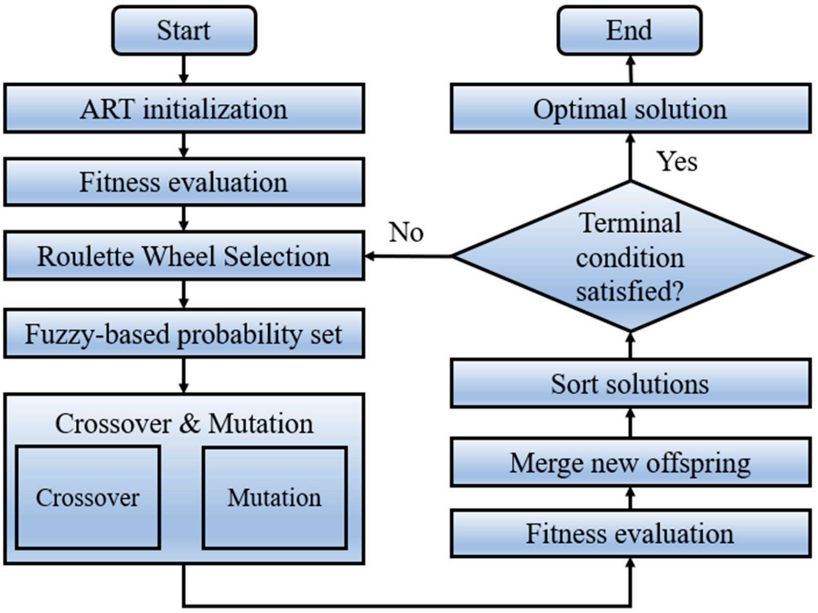

Figure 5. Flowchart of the proposed algorithm.

4. Results and Discussion

In this section, illustrative simulations have been carried out to evaluate the performance of our proposed method progressively through convergence test and simulation under a time-varying environment. The simulations are conducted via MATLAB 2021a environment with a PC configured with Intel (R) Core (TM) i7-13400 CPU and 16-GB RAM.

4.1. Convergence and Quality Test

In this section, simulations have been carried out to analyze the convergence characteristic of our proposed method. We have selected some other state-of-the-art methods from existing reliable references for comparison, including conventional genetic algorithm, D* lite [16], Hybrid A* [29], and RRT* [30]. The selected environment maps are presented in Figure 6, the start and goal points are marked as blue and red dots, respectively. It is worth noting that since we only test the convergence behavior and solution quality, the effects of time-varying currents are not considered. To maintain the efficiency and without loss of solution quality, we set the maximum number of waypoints in each path is 20 according to [31]. Table 1 shows the dimensions and coordinates of the given points.

Table 1. Environment setting.

	Map Size	Start	Destination
Scenario 1	300 × 400	(150, 300)	(200, 90)
Scenario 2	300 × 400	(50, 330)	(250, 250)
Scenario 3	400 × 400	(290, 350)	(390, 60)
Scenario 4	400 × 400	(150, 240)	(350, 120)

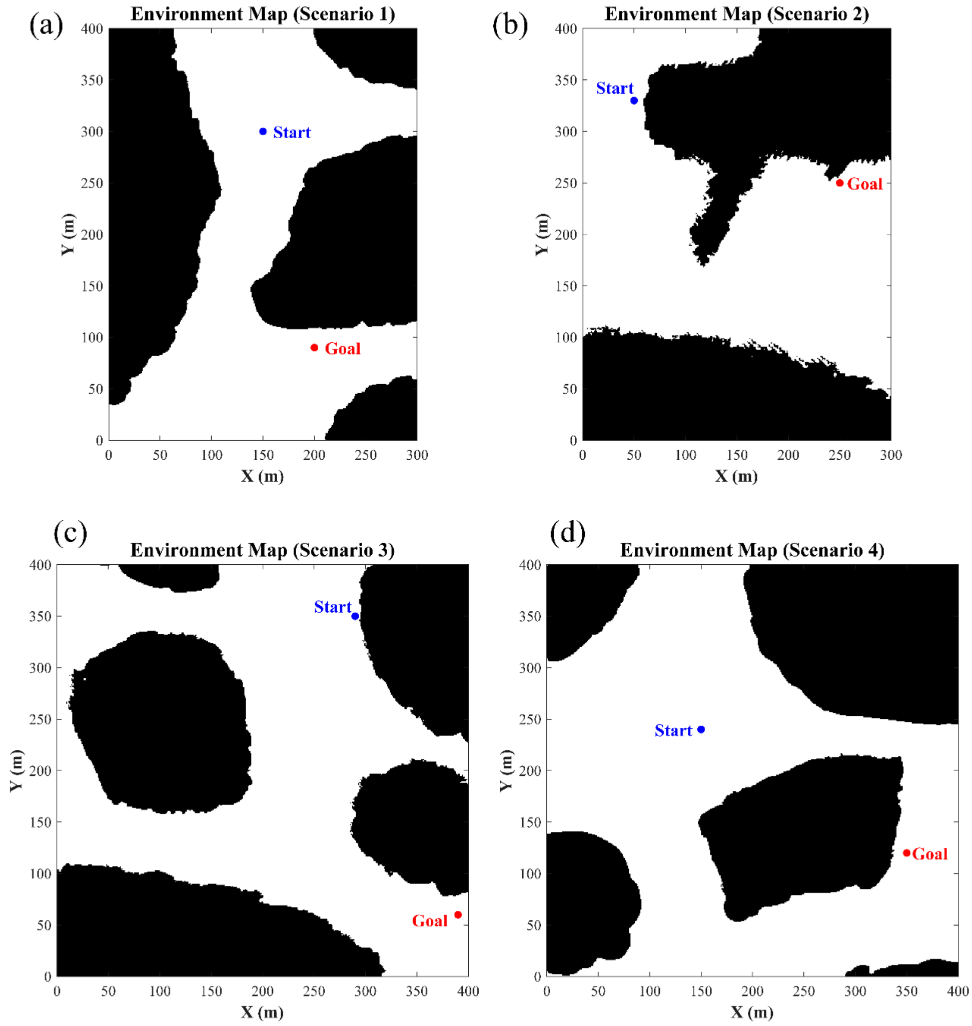
(a)
Environment Map (Scenario 1)
Start
Goal
(b)
Environment Map (Scenario 2)
Start
Goal
(c)
Environment Map (Scenario 3)
Start
Goal
(d)
Environment Map (Scenario 4)
Start
Goal
Y (m)
X (m)
0
50
100
150
200
250
300
350
400

Figure 6. Environment map (**a**) Scenario 1; (**b**) Scenario 2; (**c**) Scenario 3; (**d**) Scenario 4.

The parameters are set as follows:

- In this case, the ocean current is fixed with velocity of 0.3 m/s and direction of −70°.
- Proposed: Population size = 100; generation = 200; $p_{c_0} = 0.8$; $p_{m_0} = 0.05$; $d_{min} = 5$ m; $d_{max} = 30$ m; $w_t = 0.395$; $w_\theta = 0.275$; $w_d = 0.330$; $a = 150$; and $b = 50$.
- Traditional GA: Population size = 100; generation = 200; $p_c = 0.8$; and $p_m = 0.05$.
- D* lite: search directions = 8.
- Improved hybrid A*: Minimum turning radius = 4 m; Motion primitive length = 4 m.
- RRT*: Max-iteration = 2500; Max-Connection distance = 1 pixel.

The tabulated data presented in Table 2 provides quantitative results for the proposed algorithm. The results reveal that the algorithm exhibits a significant reduction in time cost with 0.36 s, 0.613 s, 0.484 s, and 0.391 s for the four scenarios, respectively. This represents a considerable improvement of over 60% when compared to the traditional GA. The algorithm's increased speed is primarily due to the new initialization process. Additionally, the algorithm's robustness is evaluated through the standard deviation (SD) of the time cost, which is 0.012 s, 0.022 s, 0.022 s, and 0.008 s for each case, respectively. Furthermore, the proposed algorithm is shown to provide a satisfactory minimum path length of 253.5 pixels, 352.0 pixels, 356.0 pixels, and 298.0 pixels for each scenario, respectively. Although other methods may produce slightly smaller values in some cases, the proposed algorithm provides more practical and reasonable solutions. It is important to note that the relatively low path lengths produced by D* lite and Hybrid A* are due to their reliance on optimal search based on A*, which aims for the shortest path. However, this approach often results in paths that are too close to obstacles.

Table 2. Statistical results.

Methods	Performance	Scenario 1	Scenario 2	Scenario 3	Scenario 4
Proposed	Time (s)	0.360	0.613	0.484	0.391
	Time SD (s)	0.012	0.022	0.022	0.008
	AVG length (pixels)	253.880	354.020	358.460	300.210
	Minimum length (pixels)	253.500	352.000	356.000	298.000
GA	Time (s)	0.905	1.113	1.132	1.179
	Time SD (s)	0.068	0.074	0.102	0.115
	AVG length (pixels)	273.590	373.000	368.490	395.340
	Minimum length (pixels)	251.000	348.000	353.500	296.500
D* lite	Time (s)	3.789	14.946	2.589	3.683
	Time SD (s)	0.155	0.603	0.037	0.064
	AVG length (pixels)	253.296	356.718	353.480	299.841
	Minimum length (pixels)	253.296	356.718	353.480	299.841
Hybrid A*	Time (s)	4.731	1.093	9.127	3.612
	Time SD (s)	0.096	0.035	0.376	0.065
	AVG length (pixels)	253.676	351.758	369.485	303.112
	Minimum length (pixels)	253.676	351.758	369.485	303.112
RRT*	Time (s)	0.449	1.906	1.571	3.346
	Time SD (s)	0.171	0.457	0.552	1.309
	AVG length (pixels)	324.637	487.845	497.646	458.019
	Minimum length (pixels)	287.810	403.188	413.889	358.982

The visualized results of the simulations are presented in Figure 7. The red curve depicts the smoothed path generated by the proposed method. The results demonstrate that the curve is smooth without any abrupt turns. On the other hand, the results generated by GA, D* lite, Hybrid A*, and RRT* exhibit relatively large angle changes, particularly RRT*, which exhibits the poorest performance in terms of smoothness. Furthermore, the proposed method produces a safer path than the other methods. This is primarily due to the inclusion of a safety distance term in the cost function, which forces the path to remain at a safe distance from its nearest obstacle. In contrast, the results produced by other methods often remain too close to the obstacles in some sections of the path.

Table 3 presents a comparative study of path quality, focusing on two key features: minimum clearance d from obstacles and path smoothness. The minimum clearance from obstacles measures the safety level of the results by calculating the distance between each path segment and its nearest obstacle. It is important to note that the safety distance utilized in the simulation is set at 5 m. Path smoothness measures the degree of smoothness of the path. The results presented in Table 3 reveal that the proposed method produces the safest path with the minimum distance from obstacles of 12.649 m, 11.663 m, 10.557 m, and 5.0 m for each scenario, respectively. In contrast, traditional GA and RRT* fail to satisfy the safety requirement in most cases. Moreover, the methods given by D* lite and Hybrid A* exhibit the worst performance in terms of safety, failing in all scenarios. Therefore, they are not suitable for real-world applications. In terms of path smoothness, the proposed method produces the smoothest paths (as seen in Figure 7) with values of 174.547, 149.454, 129.088, and 211.538 for each case, respectively, significantly outperforming the other methods. For instance, the proposed method's path smoothness value is 5–6 times smaller than that of Hybrid A* and D* lite.

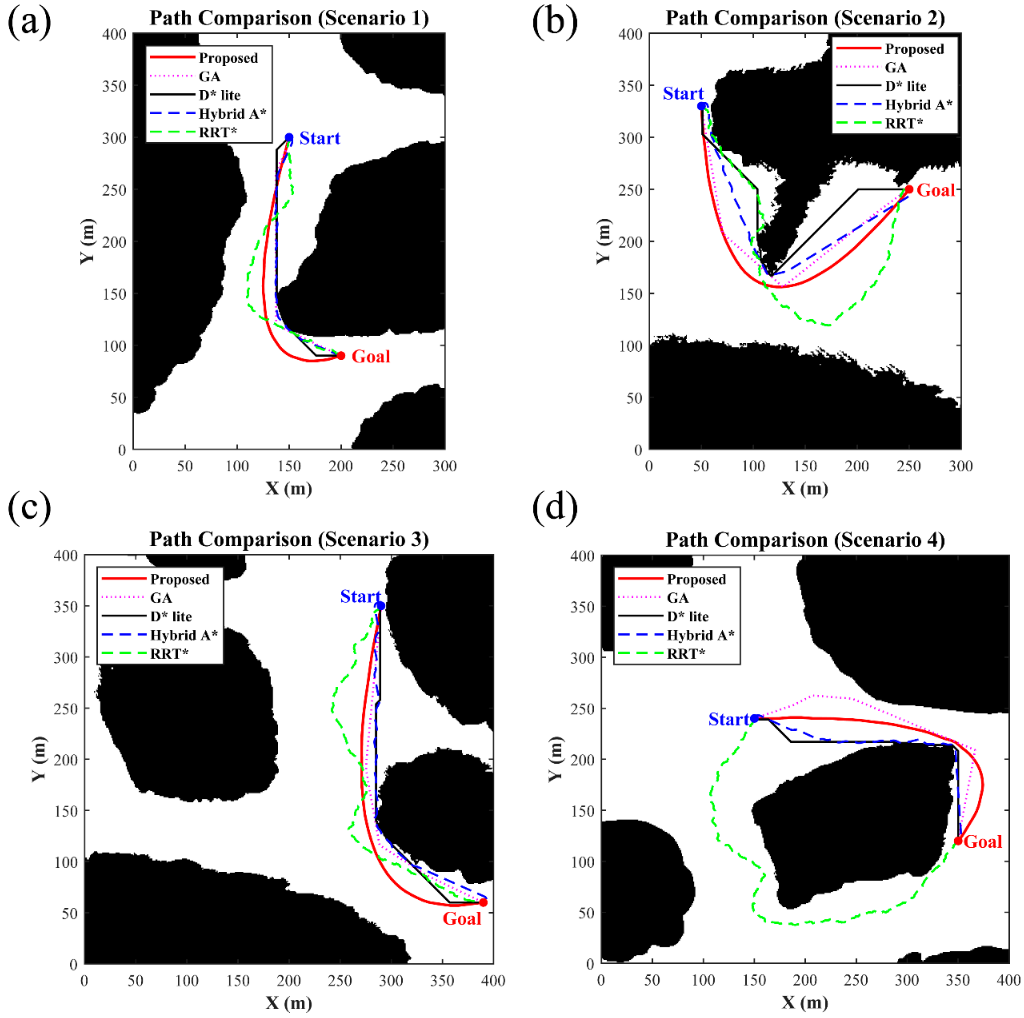

Figure 7. Visualized results (**a**) Scenario 1; (**b**) Scenario 2; (**c**) Scenario 3; (**d**) Scenario 4.

Table 3. Comparison of path quality.

Methods	Performance	Scenario 1	Scenario 2	Scenario 3	Scenario 4
Proposed	*d* [1] (m)	12.649 (○)	11.663 (○)	10.557 (○)	5.000 (○)
	Smoothness (deg)	174.547	149.454	129.088	211.538
GA	*d* (m)	1.000 (✕)	1.414 (✕)	1.414 (✕)	5.099 (○)
	Smoothness (deg)	590.291	343.702	528.768	227.737
D* lite	*d* (m)	1.000 (✕)	1.000 (✕)	1.000 (✕)	0 (✕)
	Smoothness (deg)	945	720	720	450
Hybrid A*	*d* (m)	1.000 (✕)	1.000 (✕)	0 (✕)	1.000 (✕)
	Smoothness (deg)	1223	1404	1912	1263
RRT*	*d* (m)	1.414 (✕)	1.000 (✕)	5.831(○)	2.236 (✕)
	Smoothness (deg)	303.395	555.388	426.283	762.342

[1] A safety distance of 5 m is established. If the minimum clearance requirement is met, a symbol of ○ is displayed within the bracket, while ✕ is used when it is not met.

4.2. Testing in USV Model

In this subsection, simulation studies and comprehensive comparisons are provided by conducting experiments on a prototype USV Otter (see Figure 8, www.maritimerobotics.com (accessed on 1 May 2023), Table 4 shows the maneuvering derivatives). It is worth mentioning that the paths generated by D* lite and Hybrid A* are not suitable for real application because they would collide with the obstacles. Therefore, the paths given by GA and RRT* are selected for the simulation.

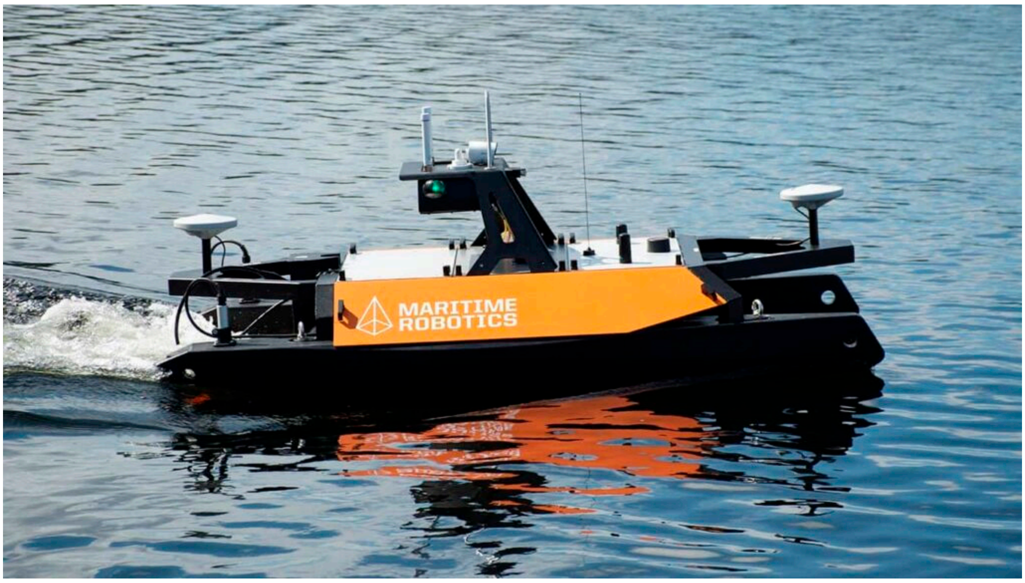

Figure 8. USV model.

Table 4. Maneuvering derivatives of Otter.

Inertial Related	Value	Damping Related	Value
m_{11}	85.28	d_{11}	−77.55
m_{22}	162.50	d_{22}	−0.02
m_{33}	41.45	d_{33}	−41.45

In simulations, Figure 9 demonstrates the alterations in course angle and speed for scenario 1 and scenario 2. The proposed method generates a path with gentle and steady changes in course and velocities, as depicted in the figures. The small deviation between the actual and desired signals suggests the feasibility of the proposed method in conjunction with the USV control system. However, the path created by GA exhibits sudden changes in both signals, resulting in a significant deviation at the point of the sudden course alteration. On the other hand, RRT* generated the poorest solution, with jagged changes in course angle and a substantial deviation between the reference and actual signals. Additionally, the simulation shows unstable velocity. Similar results are obtained by scenario 3 and scenario 4.

In Table 5, the energy and time cost of the simulations are displayed. The proposed method exhibits the lowest energy and time cost among all cases, as demonstrated in the table. This outcome is mainly attributed to the novel cost function incorporated into the proposed method, which directs the USV to move along the current direction. Conversely, the results produced by RRT* display the poorest performance with the highest energy consumption and computational time. This inferior performance is primarily due to the numerous abrupt changes in course angle and unnecessary turns.

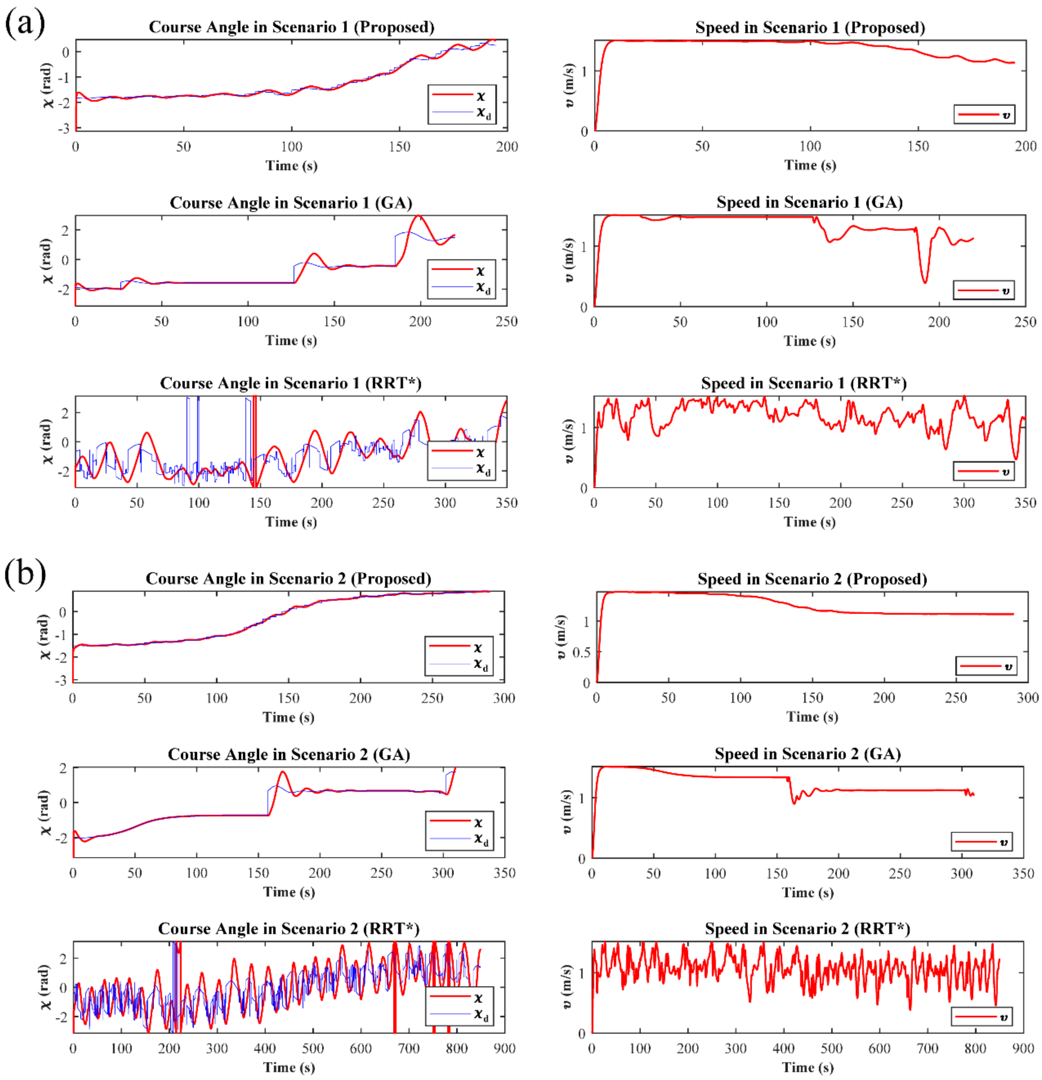

Figure 9. Simulation results for proposed method, GA, and RRT*: (**a**) Scenario 1; (**b**) Scenario 2.

Table 5. Energy cost and time cost during the mission.

	Proposed		**GA**		**RRT***	
	Energy cost (kJ)	Time cost (s)	Energy cost (kJ)	Time cost (s)	Energy cost (kJ)	Time cost (s)
Scenario 1	27.4	194.8	29.8	219.9	42.7	349.9
Scenario 2	36.8	298.9	39.1	310.0	94.0	849.6
Scenario 3	37.4	266.6	39.6	289.8	65.8	530.0
Scenario 4	31.7	246.9	35.0	285.0	82.4	708.9

4.3. Simulation in Time-Vary Ocean Environments

In this section, we will evaluate the method under time-varying ocean currents. We have selected some state-of-the-art methods from existing reliable references for comparison, including improved artificial fish swarm algorithm [8] and multi-objective enhanced GA (MOEGA) [32]. The ocean current model used in this paper is based on the numerical solution of water jet structure [33]:

$$\phi(x,y) = 1 - \tan\text{h}\left(\frac{y - B(t)\cos(k(x-ct))}{(1 + k^2 B(t)^2 \sin^2(k(x-ct)))^{\frac{1}{2}}}\right) \tag{27}$$

where $B(t)$ and k are the properly adimensionalized amplitude and wavenumber of the undulation in the stream function. The specific expression for $B(t)$ is:

$$B(t) = B_o + \epsilon\cos(\omega t + \theta) \tag{28}$$

with $B_o = 1.2$, $c = 0.12$, $k = 0.84$, $\omega = 0.4$, $\epsilon = 0.3$ and $\theta = \pi/2$. The velocity field is obtained by the following expression:

$$U(x,y,t) = -\frac{\partial\phi}{\partial y} \qquad V(x,y,t) = \frac{\partial\phi}{\partial x} \tag{29}$$

where $U(x,y,t)$ and $V(x,y,t)$ are the x– and y- components of the velocity vector at time in the location $(x,\ y)$.

The parameters are set as follows:

- Environment: map size: 500 * 500, Start = (80, 150), Goal = (480, 330), and the ocean current is set as Equation (27) for Case 1 while we multiply -1 on the y component for Case 2.
- Proposed: Population size = 100, generation = 200, $\boldsymbol{p}_{c_0} = 0.8$, $\boldsymbol{p}_{m_0} = 0.05$, $d_{min} = 5$ m, $d_{max} = 30$ m, $\text{w}_t = 0.395$, $\text{w}_\theta = 0.275$, $\text{w}_d = 0.330$, $a = 150$, and $b = 50$.
- IAFSA: Population size = 100, $\delta = 0.618$, $step = 1$; $visual = 10$, and $try\_number = 8$.
- MOEGA: Population size = 100, generation = 200, $\boldsymbol{p}_c = 0.8$, $\boldsymbol{p}_m = 0.05$, $d_{min} = 5$ m, $d_{max} = 30$ m, $\text{w}_t = 0.395$, $\text{w}_\theta = 0.275$, and $\text{w}_d = 0.330$.

Table 6 presents the quantitative outcomes, including path distance, cruising time ($\mathscr{T}$), smoothness (ϑ), and minimum distance to obstacles (d), for both Case 1 and Case 2. The visualized solutions for each case are also depicted in Figures 10 and 11. According to the results in Table 6, the proposed algorithm delivers solutions with higher quality paths (as highlighted in bold in the table), outperforming other methods in terms of cruising time, smoothness, and safety in most scenarios. However, it is noteworthy that the proposed algorithm results in the lowest safety value (8 m to the obstacle) due to the significant impact of energy consumption on optimization. It should be noted that an 8 m safety distance is acceptable in real application [34]. Moreover, the IAFSA method yields the worst outcomes, which may be attributed to its random behavior during the algorithm process, leading to abrupt points along the path. As demonstrated in Figures 10 and 11, our proposed model leverages the currents to reduce energy consumption by selecting intersection points that align with the current direction. Overall, the presented results indicate that the proposed method exhibits superior performance to the other two algorithms.

Table 6. Statistical measurements of the paths.

	Indicators	**Proposed**	**IAFSA**	**MOEGA**
Case 1	Distance (m)	458.166	471.691	483.598
	$\mathscr{T}$ (s)	209.198	214.430	221.039
	ϑ (deg)	123.130	212.378	145.872
	d (m)	18.682	5.000	16.279
Case 2	Distance (m)	540.065	543.593	566.971
	$\mathscr{T}$ (s)	245.489	254.095	257.724
	ϑ (deg)	155.034	164.404	228.242
	d (m)	8.062	18.934	17.029

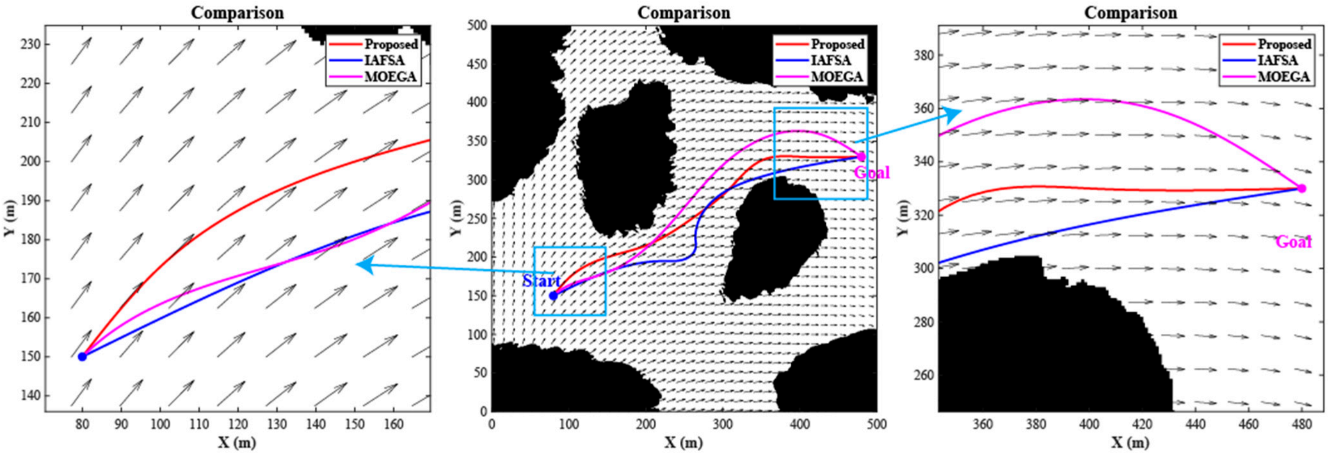

Figure 10. Visualized results of Case 1.

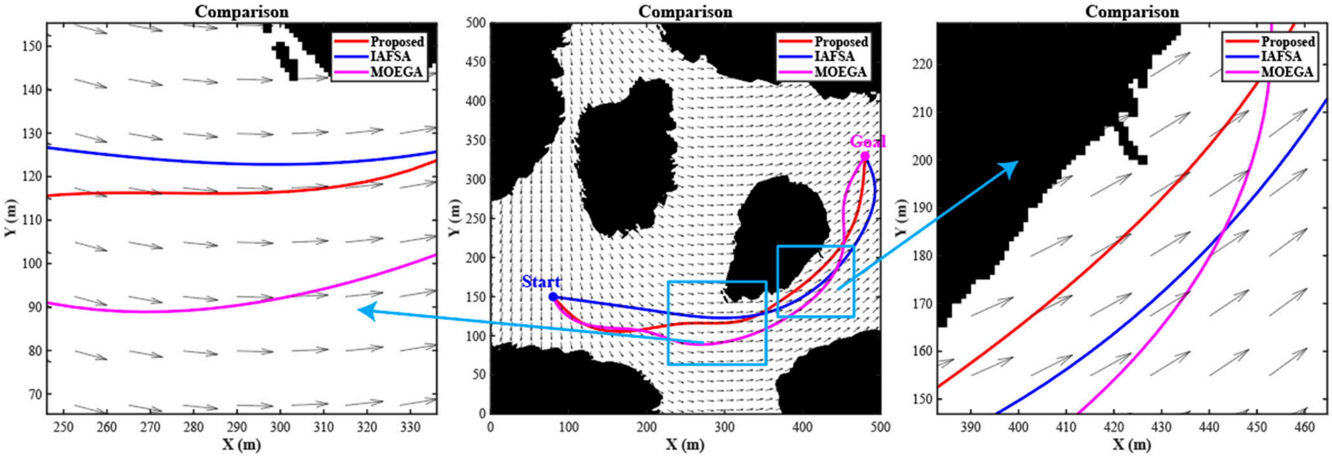

Figure 11. Visualized results of Case 2.

5. Conclusions

This paper presents a thorough investigation into the path planning problem for USVs. The proposed algorithm generates a path that is both optimally safe and quickly convergent, exhibiting strong adaptability to complex environments. In comparison to the existing literature, our method outperforms other algorithms across all problem variations. Additionally, the fast-discrete Clothoid curve is utilized to maintain path curve continuity and ensure reliable coordination between the planning and control modules, while also accommodating non-holonomic constraints. Simulation studies and comprehensive comparisons in various ocean scenarios have been conducted to illustrate the effectiveness and superiority of the proposed path planning framework.

Author Contributions: Conceptualization, F.W. and L.Z.; methodology, F.W.; software, F.W.; validation, F.W. and L.Z.; formal analysis, Y.B; writing—original draft preparation, F.W. and L.Z.; writing—review and editing, F.W. and L.Z.; visualization, F.W.; supervision, Y.B.; project administration, Y.B. and L.Z.; funding acquisition, F.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Stable Supporting Fund of Science and Technology on Underwater Vehicle Technology (grant number: JCKYS2022SXJQR-01).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: No data was used in this paper.

Acknowledgments: The authors thank the editor-in-chief, the associate editor, and the anonymous referees for their comments and suggestions.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Zhou, C.; Gu, S.; Wen, Y.; Du, Z.; Xiao, C.; Huang, L.; Zhu, M. The Review Unmanned Surface Vehicle Path Planning: Based on Multi-Modality Constraint. *Ocean Eng.* **2020**, *200*, 107043. [CrossRef]
2. Öztürk, Ü.; Akdağ, M.; Ayabakan, T. A Review of Path Planning Algorithms in Maritime Autonomous Surface Ships: Navigation Safety Perspective. *Ocean Eng.* **2022**, *251*, 111010. [CrossRef]
3. Ai, B.; Jia, M.; Xu, H.; Xu, J.; Wen, Z.; Li, B.; Zhang, D. Coverage Path Planning for Maritime Search and Rescue Using Reinforcement Learning. *Ocean Eng.* **2021**, *241*, 110098. [CrossRef]
4. Fiskin, R.; Atik, O.; Kisi, H.; Nasibov, E.; Johansen, T.A. Fuzzy Domain and Meta-Heuristic Algorithm-Based Collision Avoidance Control for Ships: Experimental Validation in Virtual and Real Environment. *Ocean Eng.* **2021**, *220*, 108502. [CrossRef]
5. Krichen, M. Improving Formal Verification and Testing Techniques for Internet of Things and Smart Cities. *Mobile Netw. Appl.* **2019**. [CrossRef]
6. Khan, M.A. A Formal Method for Privacy-preservation in Cognitive Smart Cities. *Expert Syst.* **2022**, *39*, e12855. [CrossRef]
7. Zhao, L.; Wang, F.; Bai, Y. Route Planning for Autonomous Vessels Based on Improved Artificial Fish Swarm Algorithm. *Ships Offshore Struct.* **2022**, *18*, 897–906. [CrossRef]
8. Zhao, L.; Bai, Y.; Wang, F.; Bai, J. Path Planning for Autonomous Surface Vessels Based on Improved Artificial Fish Swarm Algorithm: A Further Study. *Ships Offshore Struct.* **2022**. [CrossRef]
9. MahmoudZadeh, S.; Abbasi, A.; Yazdani, A.; Wang, H.; Liu, Y. Uninterrupted Path Planning System for Multi-USV Sampling Mission in a Cluttered Ocean Environment. *Ocean Eng.* **2022**, *254*, 111328. [CrossRef]
10. Wang, N.; Xu, H. Dynamics-Constrained Global-Local Hybrid Path Planning of an Autonomous Surface Vehicle. *IEEE Trans. Veh. Technol.* **2020**, *69*, 6928–6942. [CrossRef]
11. Yu, X.; Wang, Y. A Time Dimension-Added Multiple Obstacles Avoidance Approach for Unmanned Surface Vehicles. *Ocean Eng.* **2022**, *252*, 111201. [CrossRef]
12. Yu, J.; Liu, G.; Xu, J.; Zhao, Z.; Chen, Z.; Yang, M.; Wang, X.; Bai, Y. A Hybrid Multi-Target Path Planning Algorithm for Unmanned Cruise Ship in an Unknown Obstacle Environment. *Sensors* **2022**, *22*, 2429. [CrossRef] [PubMed]
13. Song, R.; Liu, Y.; Bucknall, R. Smoothed A* Algorithm for Practical Unmanned Surface Vehicle Path Planning. *Appl. Ocean Res.* **2019**, *83*, 9–20. [CrossRef]
14. Shah, B.C.; Gupta, S.K. Long-Distance Path Planning for Unmanned Surface Vehicles in Complex Marine Environment. *IEEE J. Ocean. Eng.* **2020**, *45*, 813–830. [CrossRef]
15. Xie, L.; Xue, S.; Zhang, J.; Zhang, M.; Tian, W.; Haugen, S. A Path Planning Approach Based on Multi-Direction A* Algorithm for Ships Navigating within Wind Farm Waters. *Ocean Eng.* **2019**, *184*, 311–322. [CrossRef]
16. Yao, Y.; Liang, X.; Li, M.; Yu, K.; Chen, Z.; Ni, C.; Teng, Y. Path Planning Method Based on D* Lite Algorithm for Unmanned Surface Vehicles in Complex Environments. *China Ocean Eng.* **2021**, *35*, 372–383. [CrossRef]
17. 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]
18. 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]
19. Hao, K.; Zhao, J.; Li, Z.; Liu, Y.; Zhao, L. Dynamic Path Planning of a Three-Dimensional Underwater AUV Based on an Adaptive Genetic Algorithm. *Ocean Eng.* **2022**, *263*, 112421. [CrossRef]
20. Krell, E.; King, S.A.; Garcia Carrillo, L.R. Autonomous Surface Vehicle Energy-Efficient and Reward-Based Path Planning Using Particle Swarm Optimization and Visibility Graphs. *Appl. Ocean Res.* **2022**, *122*, 103125. [CrossRef]
21. Xue, H. A Quasi-Reflection Based SC-PSO for Ship Path Planning with Grounding Avoidance. *Ocean Eng.* **2022**, *247*, 110772. [CrossRef]
22. Liang, C.; Zhang, X.; Han, X. Route Planning and Track Keeping Control for Ships Based on the Leader-Vertex Ant Colony and Nonlinear Feedback Algorithms. *Appl. Ocean Res.* **2020**, *101*, 102239. [CrossRef]
23. Wang, F.; Zhao, L.; Bai, Y. Path Planning For Unmanned Surface Vehicles Based On Modified Artificial Fish Swarm Algorithm With Local Optimizer. *Math. Probl. Eng.* **2022**, *2022*, 1283374. [CrossRef]
24. Ma, Y.; Hu, M.; Yan, X. Multi-Objective Path Planning for Unmanned Surface Vehicle with Currents Effects. *ISA Trans.* **2018**, *75*, 137–156. [CrossRef]
25. Wang, N.; Zhang, Y.; Ahn, C.K.; Xu, Q. Autonomous Pilot of Unmanned Surface Vehicles: Bridging Path Planning and Tracking. *IEEE Trans. Veh. Technol.* **2022**, *71*, 2358–2374. [CrossRef]
26. Meng, J.; Liu, Y.; Bucknall, R.; Guo, W.; Ji, Z. Anisotropic GPMP2: A Fast Continuous-Time Gaussian Processes Based Motion Planner for Unmanned Surface Vehicles in Environments With Ocean Currents. *IEEE Trans. Automat. Sci. Eng.* **2022**, *19*, 3914–3931. [CrossRef]

27. Liu, Y.; Bucknall, R.; Zhang, X. The Fast Marching Method Based Intelligent Navigation of an Unmanned Surface Vehicle. *Ocean Eng.* **2017**, *142*, 363–376. [CrossRef]
28. Liu, Y.; Bucknall, R. The Angle Guidance Path Planning Algorithms for Unmanned Surface Vehicle Formations by Using the Fast Marching Method. *Appl. Ocean Res.* **2016**, *59*, 327–344. [CrossRef]
29. Petereit, J.; Emter, T.; Frey, C.W.; Kopfstedt, T.; Beutel, A. Application of Hybrid A* to an Autonomous Mobile Robot for Path Planning in Unstructured Outdoor Environments. In Proceedings of the ROBOTIK 2012; 7th German Conference on Robotics, Munich, Germany, 21–22 May 2012; pp. 1–6.
30. Karaman, S.; Frazzoli, E. Sampling-Based Algorithms for Optimal Motion Planning. *Int. J. Robot. Res.* **2011**, *30*, 846–894. [CrossRef]
31. Liu, J.; Anavatti, S.; Garratt, M.; Abbass, H.A. Modified Continuous Ant Colony Optimisation for Multiple Unmanned Ground Vehicle Path Planning. *Expert Syst. Appl.* **2022**, *196*, 116605. [CrossRef]
32. Nazarahari, M.; Khanmirza, E.; Doostie, S. Multi-Objective Multi-Robot Path Planning in Continuous Environment Using an Enhanced Genetic Algorithm. *Expert Syst. Appl.* **2019**, *115*, 106–120. [CrossRef]
33. Alvarez, A.; Caiti, A.; Onken, R. Evolutionary Path Planning for Autonomous Underwater Vehicles in a Variable Ocean. *IEEE J. Ocean. Eng.* **2004**, *29*, 418–429. [CrossRef]
34. Zhao, L.; Bai, Y.; Paik, J.K. Global-Local Hierarchical Path Planning Scheme for Unmanned Surface Vehicles under Dynamically Unforeseen Environments. *Ocean. Eng.* **2023**, *280*, 114750. [CrossRef]

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.