Journal of
Marine Science and Engineering

MDPI

Review

A Review of Path Planning Methods for Marine Autonomous Surface Vehicles

Yubing Wu [1], Tao Wang [1,2,*] and Shuo Liu [1,2,*]

[1] Polytechnic Institute, Zhejiang University, Hangzhou 310015, China; 22360554@zju.edu.cn
[2] Ocean College, Zhejiang University, Zhoushan 316000, China
[*] Correspondence: twang001@zju.edu.cn (T.W.); shuoliu@zju.edu.cn (S.L.)

Abstract: A marine autonomous surface vehicle (ASV) is a kind of autonomous marine robot with intelligent and flexible use advantages. They are mainly divided into two categories: unmanned vessels and unmanned sailboats. Marine ASVs are essential in marine science, industry, environmental protection, and national defense. One of the primary challenges faced by marine ASVs is autonomously planning paths in an intricate marine environment. Numerous research findings have surfaced in recent years, including the combination with popular machine learning. However, a systematic literature review is still lacking, primarily a comprehensive comparison of two types of ASV path planning methods. This review first introduces the problem and evaluation indicators of path planning for ASVs. Then, aiming at unmanned vessels and sailboats, respectively, it sorts out various path planning algorithms proposed in the existing literature, including the advantages and limitations of both kinds of ASVs, and discusses them in combination with evaluation indicators. Also, this paper explores how marine environmental factors affect path planning and its corresponding treatment methods. Finally, this review summarizes the challenges of unmanned ship path planning, proposes potential technical solutions and future development directions, and aims to provide references for further development in this field.

Keywords: unmanned vessels; unmanned sailboats; path planning; autonomous navigation

check for updates

Citation: Wu, Y.; Wang, T.; Liu, S. A Review of Path Planning Methods for Marine Autonomous Surface Vehicles. *J. Mar. Sci. Eng.* **2024**, *12*, 833. https://doi.org/10.3390/jmse12050833

Academic Editor: Sergei Chernyi

Received: 24 April 2024
Revised: 14 May 2024
Accepted: 15 May 2024
Published: 16 May 2024

1. Introduction

Path planning, a core challenge in robotics and computer science, entails determining the most efficient route from an initial position to a target destination, while steering clear of obstacles. It is essentially about determining how to move from one point to another in a given environment, taking into account various constraints such as terrain, dynamic obstacles, and the vehicle's capabilities. In recent years, path planning has broad applications in robotics, agriculture, logistics and warehousing, land transportation, marine research, etc. In essence, path planning is a versatile concept with applications across various fields, all aimed at efficiently navigating spaces and reaching desired destinations.

As new technologies rapidly advance, ships are becoming more intelligent [1], leading to rapid development of autonomous surface vehicles (ASVs) or unmanned surface vehicles (USVs), which are gaining significance in marine science and technology [2]. An ASV is a vessel or sailboat equipped with robotic technology, capable of autonomously responding to its environment and performing tasks with minimal human intervention. It offers advantages such as long endurance, high speed, and rapid response [3]. Unlike traditional marine vehicles, ASVs can autonomously perform specialized tasks, offering energy-efficient operations, particularly for long-term missions. In recent years, researchers have found applications in civilian settings and various scientific and defense fields, including ocean data collection, marine monitoring, the detection of hostile facilities, and underwater communications [4]. Also, ASVs are used in some special fields, such as the offshore energy industry, maritime environmental monitoring and protection, search and rescue operations, etc.

A lot of professional technologies, including ship design, autonomous decision-making, control systems, data communication, and artificial intelligence, are required to develop ASVs. Ship design utilizes computational fluid dynamics (CFD) simulations like ANSYS Fluent to optimize hull shape for efficiency and stability. In autonomous surface vehicles, decision-making algorithms, such as reinforcement learning, ensure safe and efficient navigation by responding to environmental inputs like obstacle avoidance. As for control systems, like PID controllers, they adjust rudder angle or throttle based on sensor feedback to maintain desired course or speed despite external disturbances. Reliable data communication systems, including long-range wireless protocols like LTE or satellite, enable sensor data transmission and information exchange with remote operators or other vehicles. And artificial intelligence (AI) technologies, particularly CNNs, aid in object detection and route optimization, enhancing autonomous decision-making by detecting and classifying objects from sensor feeds. As a core technology of ASVs, path planning entails discovering routes that are both collision-free and efficient from the starting point to the destination [5]. Specifically, coverage path planning for ASVs aims to find routes that enable the efficient traversal of mission areas to fulfill requirements in different mission backgrounds. Path planning is crucial in marine mining, mapping, resource exploration, surface garbage removal, etc. [6].

Over the last five years, numerous review papers have emerged, offering comprehensive summaries of progress in path planning research. Below are the most pertinent ones: In research [7], an overview of recent advancements and innovative approaches in ASV path planning and obstacle avoidance techniques is provided. It also compares the constraints and environmental influences on ASVs between global and local path planning algorithms. Reference [8] provides a comprehensive review of ASV development, covering aspects such as trajectory tracking, target tracking, cooperative formation control, and path tracking. While emphasizing intelligent motion control, it provides less detailed path planning coverage. In research [9], the local path planning issue for USVs is discussed, outlining the features of different algorithms across two levels, which are trajectory optimization and path search. Research [10] delineates the progress in ASV path planning research, with a focus on multi-modality constraints, classified into three stages, which are trajectory planning, route planning, and motion planning. In recent years, numerous research findings have surfaced, as mentioned above. However, there remains a gap in systematic literature reviews, particularly concerning comprehensive comparisons between two types of ASV path planning methods: path planning algorithms for unmanned vessels and unmanned sailboats.

In this review, path planning algorithms for ASVs are summarized and discussed. The first section introduces the definition of path planning and ASVs, path planning for ASVs, and the existing review literature in this field. The second section illustrates how the path planning workspace is modeled, and indicators used to evaluate ASV path planning are demonstrated. Following this, Section 3 presents a comprehensive list of path planning algorithms for ASVs, including the categories of ASVs, path planning algorithms for unmanned vessels, path planning algorithms for unmanned sailboats, and path planning algorithms that consider marine environmental factors. In Section 4, existing research problems are summarized, and the future research direction of this field is discussed. Finally, a conclusion of this review is made in Section 5.

2. Path Planning Workspace Modeling

Path planning is fundamental to ASVs, identifying safe and efficient routes from start to destination [5]. In coverage path planning, the goal is to navigate mission areas effectively to meet various objectives in different mission contexts. Path planning for ASVs is critical in ensuring their safe navigation within working environments, directly impacting their safety and economic efficiency [9]. It is a key technology driving the automation and intelligence of ASVs, enabling them to execute complex tasks with precision and effectiveness [11].

To calculate paths effectively, an ASV path planner requires environmental information, such as obstacle presence, the machine's location to be inspected, and surface features [12]. This section introduces how the path planning workspace is modeled, mainly how the map for ASV path planning is obtained.

2.1. Different Methods of Representing the Environment

Several methods exist to construct a graph representing the environment in which the ASV is moving. As indicated by the authors of [13], multiple methods exist for constructing graphs, as depicted in Figure 1. The study in [14] has also contributed to this classification. According to the researchers, there are two types of methods for constructing a map: cell decomposition and roadmap graphs. In cell decomposition, cells can be arranged using regular or irregular grids. Figure 1a–c show how to build a map with the help of polynomials such as squares, triangles, and hexagons. According to [15], the main advantage of using regular grids is that each node can be easily indexed, which means a quick access to each node and their optimized memory storage. As for irregular grids, shown in Figure 1d, it allows for improved grid adaptation to varying terrain features by adjusting resolution, although it may result in suboptimal paths [16]. Another way of representing the environment is roadmap graphs. The graph consists of interconnected nodes, where each node denotes a potential robot state, and each edge represents a transition between states. Figure 1e,f show examples of roadmaps, including Voronoi graphs [17] displayed in Figure 1e and state lattice graphs displayed in Figure 1f. Voronoi graphs are created using Voronoi diagrams, which segment a space into areas according to proximity to specific points. In the context of robotics, these points typically represent obstacles or other key features in the environment. The state lattice graphs are a type of roadmap graph that discretizes the robot's state space into a grid of possible configurations or states.

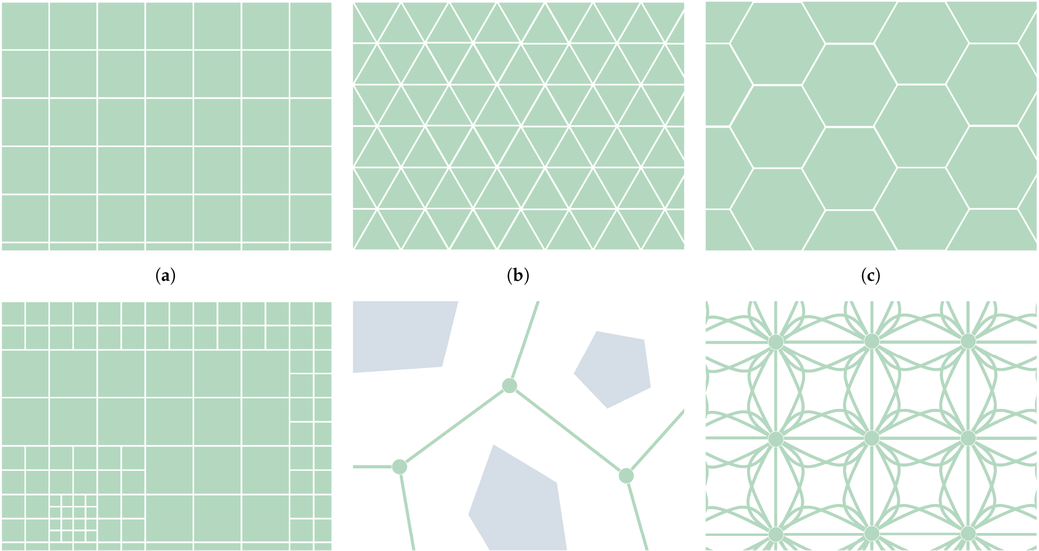

(**a**) (**b**) (**c**)
(**d**) (**e**) (**f**)

Figure 1. Different environmental unit decomposition methods (**a**–**d**) and road maps (**e**,**f**). (**a**) Square tessellations; (**b**) triangular tessellations; (**c**) hexagonal tessellations; (**d**) irregular tessellations; (**e**) Voronoi road map; (**f**) state lattice road map [12].

2.2. Static and Dynamic Maps

According to [18], path planning methodologies can be categorized as deliberative or reactive architectures based on the availability of environmental information. The former, referred to as off-line or global path planning, directs the ASVs to its end point within a mapped environment containing static obstacles such as shallow water, islands, and beacons. In [19], an ARC-Theta* algorithm is introduced, considering vehicle capabilities on an Euclidean group SE(2)-weighted occupancy grid map, to generate real-time paths. Simulation and experimental verification confirm the efficacy of the algorithm, demonstrating its effectiveness for UUV global path planning. Additionally, in [20], an extended ARC-Theta* path planning algorithm is introduced to enhance the removal efficiency of jellyfish by maintaining the JEROS system formation.

Online or local path modification or planning, the latter option, entails immediate responses to uncertainties and unexpected obstacles in critical scenarios, allowing for rapid reactions in real-time. In this category, the maps are represented as dynamic. In [21], an interactive trajectory tracking control (fu-ittc) scheme based on a finite-time unknown observer is proposed for the asymmetric underdriven surface vehicle (AUSV). Extensive experimentation and comparison conducted on a benchmark AUSV demonstrate the remarkable steady-state and transient accuracy of tracking of the FUO-ITTC method. In [22], the obstacle avoidance problem of underpowered unmanned ships under unknown environmental interference is studied. Using the robust deep q network structure, a simple obstacle avoidance algorithm of deep reinforcement learning is proposed, which resolves the usability issues stemming from the complex control laws found in traditional analysis methods. With the same CDRLOA system, goals and constraints can be extensively customized to fulfill diverse special requirements. Many experiments prove the simplicity and effectiveness of the algorithm. In [23], a ship decision support system is presented, designed to address collision avoidance challenges posed by obstacles which are static and dynamic. This system translates the decision-making capabilities of human navigation experts into solving path planning issues for ships navigating complex environments. Further development can be made to enable automatic ship control capabilities. The results demonstrate the effectiveness of the proposed method in resolving ship path planning challenges by considering static and dynamic obstacles in the environment, maritime traffic rules, and ship dynamic characteristics, making the method suitable for commercial systems.

In real scenarios, path planning algorithms for ASVs must consider marine environmental factors like ocean currents, wind, waves, tides, infrastructure, and weather conditions, which can impact their outcomes. Addressing these factors is also crucial for ensuring the safety and efficacy of path planning processes.

2.3. Path Planning Evaluation Indicators

To choose the best path for an ASV to travel, several indicators will be used to evaluate different algorithms. Some indicators are often used in ASV path planning evaluation, including time of path planning, total simulation time, distance traveled, energy consumption, and security level.

The time of path planning is the duration required for an algorithm to produce a new path. It records the time from when the vehicle initiates path planning until the completion of calculations and acquisition of a valid path. Regarding distance traveled, it represents the overall distance required to traverse from the initial point to the destination of each algorithm's resulting path. Moreover, energy consumption is the total energy expenditure needed to travel from the start point to the destination following the planned path, which is positively correlated with the distance traveled. Moreover, the security level assesses the level of safety or the degree of security in executing the planned path [24].

3. Path Planning Algorithms for ASVs

3.1. Classification of ASV

According to [25], ASV can be categorized into two types: unmanned vessels and sailboats. Figure 2 shows the examples of these two kinds of ASVs.

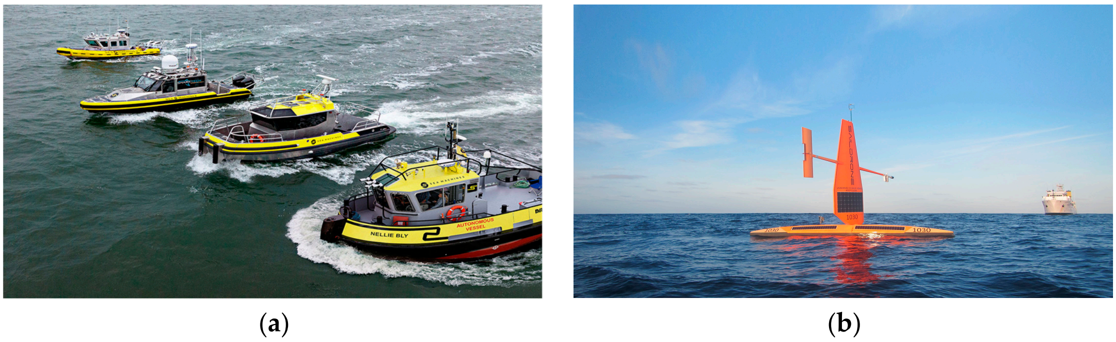

(**a**) (**b**)

Figure 2. Examples of unmanned vessels and unmanned sailboats. (**a**) An example of unmanned vessels [26]. (**b**) An example of unmanned sailboats [27].

In recent years, unmanned vessels have attracted significant attention because they have several advantages over unmanned sailboats. Initially, they employ propulsion systems equipped with robust propellers rather than depending on unpredictable wind. Secondly, these vessels provide a versatile array of turning options and higher speeds compared to sailboats, thereby improving efficiency and appeal. Finally, unmanned vessels serve diverse and valuable purposes across marine transportation, encompassing cargo vessels, naval ships, merchant ships, yachts, and passenger boats [4].

Unmanned sailboats pose greater complexity compared to unmanned vessels due to the additional external factors they must account for. Although the path planning of unmanned sailboats is similar to that of unmanned vessels, sailboats rely solely on sail power rather than steady propulsion from propellers. Due to the unpredictable winds, ocean waves, and currents, the direction and velocity of sailboats are influenced, which drives the operations of unmanned sailboats within an unpredictable ocean environment. Among these impact factors, the only power that keeps sailboats sailing is the wind, meaning less energy is consumed than unmanned vessels. The low power consumption and enduring operational cycles enable unmanned sailboats to independently accomplish specialized tasks, particularly long-term and uninterrupted ocean tasks [4]. There are several differences between these two kinds of ASVs. The advantages of unmanned vessels are robust resistance to environmental interference and broad applicability. However, fuel consumption is a disadvantage for them. As for unmanned sailboats, they have advantages such as no fuel consumption, energy efficiency, no propel system, and a simpler structure. Meanwhile, they have the disadvantage of being prone to environmental factors.

Both types of ASVs, however, do have examples of operating on solar power. These ASVs face a challenge when operating at night since they rely on sunlight to generate power through solar panels. To solve this problem, some ASVs are equipped with energy storage systems, typically batteries, which store excess energy generated during daylight hours. These batteries then power the vehicle during periods when sunlight is unavailable, such as at night [28].

3.2. Path Planning Algorithms for Unmanned Vessels

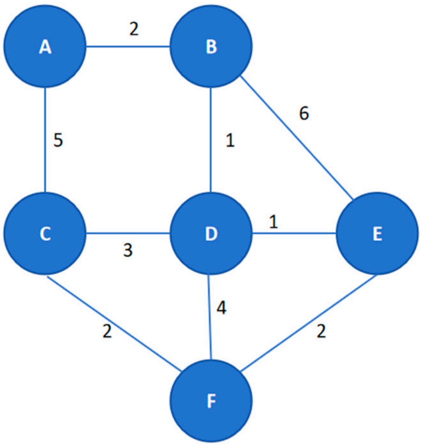
In recent years, researchers have found several path planning algorithms for unmanned vessels, categorized as graph-search-based algorithms, randomization methods based on optimization techniques, dynamic path generation-based algorithms, and machine learning-based algorithms.

3.2.1. Graph-Search-Based Algorithms

Graph-search-based algorithms serve as the path planning method for unmanned vessels, utilizing a graph for navigation. In this discretization of space, nodes represent the vessel's starting point, end points, and all potential reachable locations. Links, or edges, indicate possible travel paths between nodes for a material point on the graph. Classical algorithms like the Dijkstra algorithm and the A* algorithm fall into this category.

1. Dijkstra Algorithm

The Dijkstra algorithm [29] is a breadth-first search with priority definition. An example of how the traditional Dijkstra algorithm works is displayed in Figure 3. Firstly, identify and list the points within the environmental road network near the initial source point that have not yet been processed. Place the points that satisfy the criteria into the START array for further evaluation. Next, choose the point closest to the initial source point from the START array and transfer it to the FINISH array. After that, explore each current point's sub-nodes, compute the distances from the starting source point to these sub-nodes, and store these distances in an array for sorting, including the sub-nodes back in the START array. Continue performing the previous two steps until the START array is empty or the target point has been reached. When dealing with a matrix environment, such as a grid-based map, the Dijkstra algorithm can still be applied for path planning by converting the grid into a graph representation. Every grid cell symbolizes a node within the graph. The nodes are connected to their neighboring cells (up, down, left, right, and optionally diagonally), forming edges between them. After the transformation, the Dijkstra algorithm can be applied following the steps mentioned above.



Step1. START array：[A (0)] FINISH array：[]
Step2. START array：[] FINISH array：[A (0)]
Step3. START array：[B (2), C (5)] FINISH array：[A (0)]
Step4. START array：[C (5), D (3), E (8)] FINISH array：[A (0), B (2)]
Step5. START array：[C (5), E (8)] FINISH array：[A (0), B (2), D (3)]
Step6. START array：[C (5), E (4), F (7)] FINISH array：[A (0), B (2), D (3)]
Step7. START array：[C (5), E (4)] FINISH array：[A (0), B (2), D (3)]
Step8. START array：[C (5), F (7)] FINISH array：[A (0), B (2), D (3), E (4)]
Step9. START array：[C (5), F (7)] FINISH array：[A (0), B (2), D (3), E (4)]
Step10. START array：[C (5)] FINISH array：[A (0), B (2), D (3), E (4), F (7)]
Step11. START array：[] FINISH array：[A (0), B (2), D (3), E (4), F (7)]

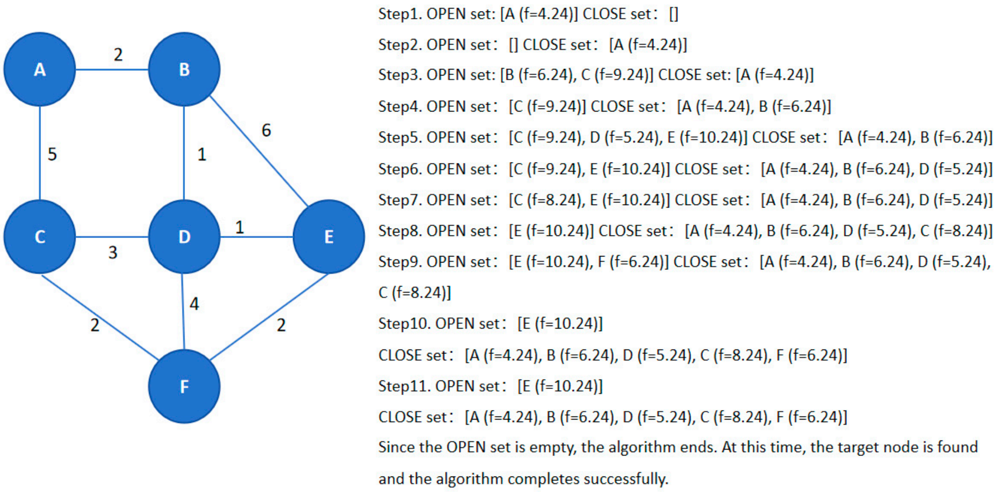
Figure 3. An example of traditional Dijkstra algorithm. Nodes A-F stand for points within the environmental road network, numbers on the line stand for distances between every two points.

An inflection point marks a spot on a curve where the curvature shifts its direction. Put plainly, it is where the curve transitions from concave to convex, or the other way around. Essentially, at an inflection point, the curve alters its curvature's orientation. In [30], the original Dijkstra algorithm is employed for path planning, but it often results in numerous unnecessary inflection points which generate a longer path. To tackle this problem, the concept of pheromones from the ant colony algorithm is incorporated into the criteria of the classic Dijkstra algorithm. Experimental results demonstrate that this optimized approach significantly reduces excrescent points in the process of path planning.

The Dijkstra algorithm uses breadth-first research without a heuristic function. It is less efficient due to exhaustive research, which always causes comparatively long path planning times. Also, it always generates unnecessary inflection points, which causes longer distance traveled and energy consumption.

2. A* Algorithm

The A* algorithm [31] is another heuristic search method based on priority definitions. It identifies the optimal path within a connected graph by utilizing a heuristic evaluation function. When selecting the next exploration node, it assesses the node using the heuristic evaluation function and chooses the node with the lowest path cost as the subsequent exploration node to proceed to. An example of how the A* algorithm works is displayed in Figure 4.



Step1. OPEN set: [A (f=4.24)] CLOSE set： []

Step2. OPEN set： [] CLOSE set： [A (f=4.24)]

Step3. OPEN set: [B (f=6.24), C (f=9.24)] CLOSE set: [A (f=4.24)]

Step4. OPEN set： [C (f=9.24)] CLOSE set： [A (f=4.24), B (f=6.24)]

Step5. OPEN set： [C (f=9.24), D (f=5.24), E (f=10.24)] CLOSE set： [A (f=4.24), B (f=6.24)]

Step6. OPEN set： [C (f=9.24), E (f=10.24)] CLOSE set： [A (f=4.24), B (f=6.24), D (f=5.24)]

Step7. OPEN set： [C (f=8.24), E (f=10.24)] CLOSE set： [A (f=4.24), B (f=6.24), D (f=5.24)]

Step8. OPEN set： [E (f=10.24)] CLOSE set： [A (f=4.24), B (f=6.24), D (f=5.24), C (f=8.24)]

Step9. OPEN set： [E (f=10.24), F (f=6.24)] CLOSE set： [A (f=4.24), B (f=6.24), D (f=5.24), C (f=8.24)]

Step10. OPEN set： [E (f=10.24)]

CLOSE set： [A (f=4.24), B (f=6.24), D (f=5.24), C (f=8.24), F (f=6.24)]

Step11. OPEN set： [E (f=10.24)]

CLOSE set： [A (f=4.24), B (f=6.24), D (f=5.24), C (f=8.24), F (f=6.24)]

Since the OPEN set is empty, the algorithm ends. At this time, the target node is found and the algorithm completes successfully.

Figure 4. An example of A* algorithm.

Researchers expand upon implementing the proposed A* approach to navigate through environments with static and dynamic obstacles, along with fluctuating current intensities. The effectiveness of the proposed approach is evaluated through simulations under various environmental conditions, focusing on path length and computational time. The findings indicate the viability of the proposed approach for the global path planning of USVs [32].

In [33], the A* algorithm is utilized for global path planning based on a model, whereas the artificial potential field (APF) is applied for local path planning in unknown areas. Furthermore, an A* algorithm with self-learning capabilities is introduced for secondary planning and situations where a local path cannot be produced. Simulation results indicate that the proposed algorithm surpasses other algorithms in terms of distance and time costs.

In [34], the research leverages the tendency for optimal ship paths to only alter course at obstacle corners. Introducing an enhanced A* algorithm, ABG, proves more efficient for autonomous inland waterway ships. Simulation experiments, conducted at two accident-prone locations, compare the performance of A, APS, Theta, and ABG. Considering both path length and computational time, ABG demonstrates superior performance in inland autonomous ship path planning.

In [35], taking into account the maneuvering traits of unmanned surface vehicles, an enhancement of the commonly employed A* algorithm is introduced, termed the Label-A* algorithm. This advancement aims to address the motion planning challenges of unmanned surface vehicles navigating restricted waters more effectively. To validate both the established model and the proposed algorithm, numerical simulations and field tests are conducted. The experimental outcomes demonstrate that the motion planning method can effectively meet the state constraint, maneuvering characteristic constraint, and surface scale constraint of unmanned surface vehicles simultaneously.

Additionally, [36] presents an optimal path planning technique tailored for unmanned surface ships, taking into account real marine environments. Diverging from traditional open/barrier classification methods, this approach introduces an enhanced risk penalty-associated A* algorithm. By incorporating new obstacle modeling techniques, risk penalties are defined for potential hazard zones based on ship weight and ocean weather conditions. Ultimately, the effectiveness of the proposed method is verified using a virtual engine.

Moreover, a hybrid path planning algorithm combining A* and DWA algorithms is proposed in [37]. The unmanned surface ship (USV) iteratively updates the local target points obtained by the A* algorithm and uses DWA to obtain the best search path. The simulation outcomes indicate that, in comparison with the A* algorithm, the proposed algorithm reduces path length by 21% and path inflection points by 59%, underscoring its superior performance.

Additionally, upon addressing concerns such as excessive memory usage, extended computation time, and an increased likelihood of encountering "dead zones", an enhanced A* algorithm is introduced for unmanned ship path planning in [38]. The grid method is selected to build the unmanned ship's driving environment model. A* algorithm determines the cost function, the cost size is judged, and the node with the least cost is taken as the next trajectory point, thus obtaining the optimal driving path of the unmanned ship. To enhance the A* algorithm, improvements are made to the actual cost function by incorporating factors such as the minimum turning radius, minimum path length, and constraints on the traveling speed of unmanned ships. Additionally, the heuristic function is refined through a linear combination of Euclidean and Manhattan distances, streamlining the sailing path of unmanned ships, eliminating extraneous nodes, and obtaining the optimal path.

The A* algorithm utilizes a heuristic search strategy that combines g-score and h-score, as shown in Equation (1). Each adjacent cell to the current one is evaluated based on its f(n), which merges g(n) and h(n). Here, g(n) represents the path length from the initial state to the goal state through the specified sequence of cells, while h(n) indicates the heuristic distance of the cell to the goal state. The cell with the lowest f(n) value is selected as the subsequent cell in the sequence. Thanks to its heuristic guidance, the A* algorithm demonstrates enhanced efficiency in path planning time and travel distance for unmanned vessels, consequently resulting in comparatively lower energy consumption.

$$\mathrm{f}(n) = \mathrm{g}(n) + \mathrm{h}(n) \tag{1}$$

Both algorithms fall into the category of graph-search-based algorithms, but they still have some differences in the path planning of unmanned vessels, as displayed in Table 1.

Table 1. Differences between Dijkstra algorithm and A* algorithm in unmanned vessels' path planning.

Aspect	Dijkstra Algorithm	A* Algorithm
Search Strategy	Breadth-first search	Heuristic search combining g-score and h-score
Heuristic Use	No heuristic function	Utilizes a heuristic function (h-score)
Efficiency	Less efficient due to exhaustive search	More efficient due to heuristic guidance

3.2.2. Randomization Method Based on Optimization Technique

The randomization method based on optimization technique is an approach in optimization algorithms where randomness or probabilistic methods are employed to enhance the search process for optimal solutions. This method introduces random elements into the optimization process, such as randomly initializing solutions, introducing random perturbations or mutations during optimization iterations, or incorporating probabilistic decision-making mechanisms. Randomization helps prevent entrapment in local optima and facilitates a more effective exploration of the search space, potentially resulting in improved solutions to optimization problems. The ant colony algorithm and the particle swarm algorithm are examples of randomization methods based on optimization technique, which are discussed below.

1. Ant Colony Algorithm

Ant colony optimization [39] is based on simulating the foraging behavior of ants to derive optimal paths. Research has shown that when ants forage for food, they secrete a chemical known as pheromone. The amount of this pheromone inversely correlates with the distance of their route. As ants follow a trail, they detect the pheromone left by preceding ants and are drawn toward areas with higher concentrations. More pheromones are deposited along the trail, creating a positive feedback loop. When certain ants discover a shorter route, they draw additional ants to this path. Ants can identify the most direct route from their nest to the food source through ongoing repetition. Initially, the pheromone level at each node is uniform. Each ant, referred to as ant k, decides its next move based on the pheromone levels at each node. In ant colony optimization, each ant's journey represents a potential solution to the optimization problem. The combined paths of all ants form the solution space, and the path chosen by the entire ant population represents the optimal solution. Figure 5 shows how an ant colony algorithm works. With an ant colony algorithm, the shortest path between two points, A and B, emerge from the combination of several ants' choices. In Figure 5, the shortest path between A and B is the path highlighted in gray.

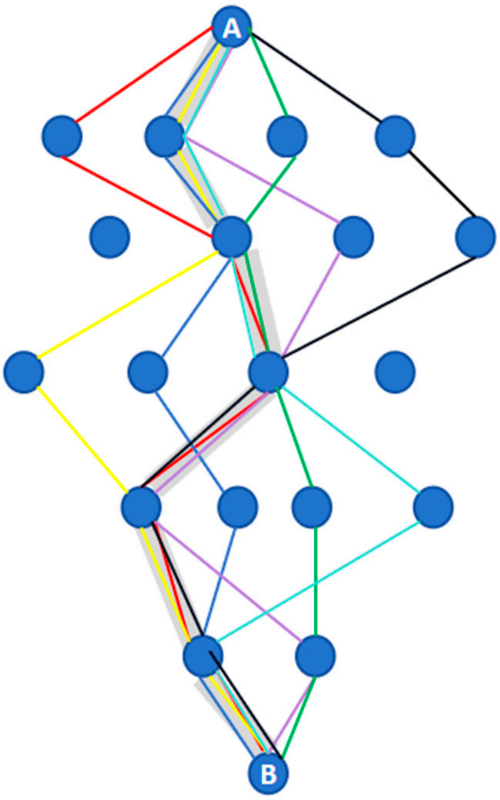


Figure 5. An example of ant colony algorithm. A is the start point and B is the end point. Paths in different colors stand for different ants' path choices.

In [30], based on the optimal path, the pheromone element of ant colony optimization is added for exclusion, and the measured result is put into the FINISH table after deletion so that the final value can be traced from the end point to the start point. Furthermore, experimental results indicate that the optimized algorithm notably decreases redundant points in path planning and reduces the routing cost of unmanned ships.

In [40], an enhanced ant colony algorithm is proposed to address challenges encountered in the practical application of unmanned ships. Introducing a novel heuristic factor, it dynamically adjusts the pheromone volatilization coefficient by leveraging Gaussian function characteristics to control unmanned ship movement direction, thereby efficiently reducing the time taken for unmanned ships to reach target points. Simulation results demonstrate that, compared to the traditional ant colony algorithm, the enhanced version optimizes the shortest path and pathfinding time for UUVs while enhancing the smoothness of their search paths.

In [41], an enhanced ant colony clustering algorithm is introduced, offering an automatic selection of an appropriate search range through matching with different environmental complexities. This feature optimizes the utilization of ASV's limited computing resources while enhancing path planning performance. Additionally, ASV maneuvering rules and a smoothing mechanism are employed to adjust and refine the dynamic search path, effectively reducing path length and cumulative angle. A simulation example illustrates the algorithm's ability to identify suitable obstacle distributions within the search range for various targets, showcasing its adaptability in completing path planning tasks.

Furthermore, leveraging the ant colony algorithm, local path planning is conducted for a simplified model of ground unmanned vehicles to achieve static collision avoidance [42]. Adhering to the International Convention on Regulations for Preventing Collisions at Sea (COLREGS), a dynamic obstacle model is formulated and computed to determine speed and course adjustments of unmanned surface vehicles across various encounter scenarios.

Moreover, addressing the challenge of initial search inefficiency in the ant colony algorithm, a hybrid approach is proposed by integrating it with the particle swarm optimization algorithm [43]. This combined algorithm addresses the mathematical model for autonomous navigation path planning of unmanned ships. Experimental results demonstrate the superior path planning accuracy of the proposed method, with improvements observed in path length, average energy consumption, and path planning time indices.

In [44], a path planning algorithm leveraging an enhanced ant colony approach is introduced. Path transfer probabilities are comprehensively calculated using an improved heuristic function that considers ship turning angles and collision risks. Additionally, an initial pheromone concentration allocation strategy based on distance factors is proposed to enhance the algorithm's early search efficiency. Simultaneously, an adaptive pheromone volatilization coefficient is suggested to further enhance the algorithm's initial convergence speed and global search capability.

To address the global path planning challenge for surface unmanned ships, Ref. [45] introduces the Bacterial Foraging-Improved Ant Colony Algorithm (BF-IACA). Unlike the traditional ant colony algorithm, the proposed algorithm's path search strategy considers constraints such as minimizing the number of turns and completely avoiding excessive turning angles. It incorporates a steering angle heuristic factor to comprehensively address transition probability. Furthermore, the reproduction and chemotactic operations of the bacterial foraging algorithm are integrated to enhance the pheromone updating mechanism, overcoming the traditional ant colony algorithm's tendency to converge slowly and become entrapped in local optima.

The ant colony algorithm utilizes pheromone trails to build solutions and communicates through indirect pheromone sharing. Its implementation is simpler, and fewer parameters are involved. However, it has the disadvantages of local optimization and too slow a convergence speed. Therefore, the time of path planning is relatively long.

2. Particle Swarm Algorithm

The particle swarm algorithm [46] is an intelligent bionic approach inspired by the foraging behavior of birds. It achieves optimal solutions through random exploration and information exchange among individuals in a population. Initially, birds are dispersed randomly across search areas with only the knowledge of the presence of food, but not its exact location. By gauging the distance to food from their respective locations, individuals assist one another in guiding the birds toward the optimal individual, eventually congregating near the food source. In the particle swarm optimization algorithm, the solution to the problem is represented by the position of individual particles within the swarm. Information exchange among particles guides their movement, leading to the identification of the best solution for the problem. In the particle swarm algorithm, a group of m particles navigates through an n-dimensional search space. Figure 6 illustrates the velocity update and movement process of the particles. These movements are influenced by factors such as current motion, particle memory, and swarm behavior.

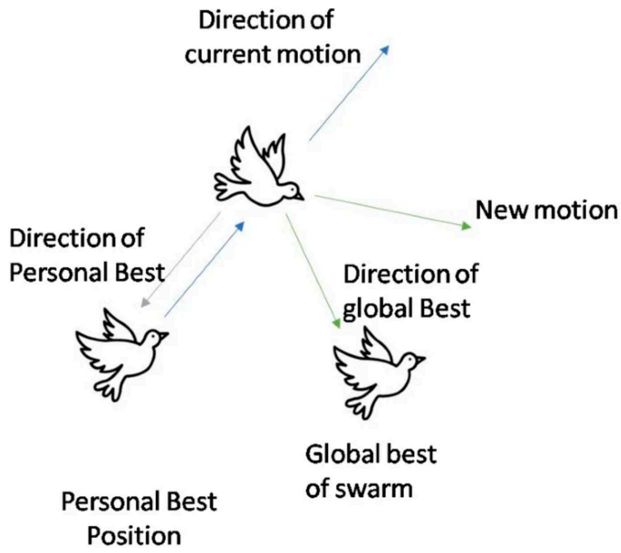


Figure 6. Schematic representation of velocity update and movement by particles [47].

This method can describe the problem to be solved in D-dimensions as shown in Equation (2), where f (x) refers to particle fitness function, $X_i = [x_{i1}\ x_{i2},\ldots,\ x_{id}\ ]$ stands for the position of the i particle, x_{dmax} and x_{dmin} stand for the upper and lower boundaries of the search space dimensions:

$$\begin{gathered}\min \mathrm{f}(x) = \mathrm{f}(x_1,x_2,\ldots x_d) \\ \mathrm{s.t.}x_{dmin} \leq x_i \leq x_{dmax},\ i = 1, 2, \ldots \mathrm{D}\end{gathered} \tag{2}$$

To address the issue of particle swarm optimization's tendency to converge to local optima, chaos theory is integrated into the basic algorithm to introduce chaotic populations. Additionally, contemporary global optimal chaos iteration replaces some particles stuck in local optima, enhancing particle diversity in later stages of population search. To further bolster the algorithm's local search capability, the particle swarm algorithm is merged with the bee swarm algorithm. This improved approach is applied to global path planning, and simulation verifies its advantages in convergence speed and search accuracy [48].

In [49], a practical and efficient path planning algorithm is presented for unmanned surface vehicles, utilizing a traditional particle swarm optimization (PSO) framework. The introduction of a grouping strategy based on preprocessing direction angles enhances computational efficiency and adaptability to the distribution characteristics of planning points. Additionally, mutation is incorporated during initialization to enhance the particle swarm's global search capability, while multiparticle competition is introduced in subsequent iterations to preserve population diversity. Once all subdomain paths are concatenated, a simplified 4-opt method is applied to further enhance and expedite convergence. In the 2-opt optimization technique, pairs of edges in a tour (a sequence of points that forms a closed loop) are swapped in order to reduce the total length of the tour. This is performed iteratively until no further improvement can be made. The name "2-opt" comes from the fact that it considers the optimization of pairs of edges. The 4-opt optimization extends the idea of the 2-opt by considering larger sets of edges to swap. Instead of just swapping pairs of edges, it looks for opportunities to swap sets of four edges to further optimize the tour.

In [50], leveraging the design and implementation context of the developed FR unmanned ship, the particle swarm optimization algorithm is employed to enhance and address challenges related to local optimization, limited adaptability to interference, and low control accuracy encountered with traditional route planning algorithms. Calculation and testing outcomes demonstrate that FR unmanned ships, by utilizing improved particle swarm optimization algorithms, are capable of adeptly navigating complex environments.

They can effectively evade surface obstacles, escape from local minima, achieve precise target reaching, converge swiftly, and meet timely response requirements.

In [51], an enhanced particle swarm optimization algorithm is introduced, integrating crossover and mutation operations from genetic algorithms into the traditional particle swarm optimization approach. Initially, the particle swarm undergoes initialization, with particle fitness being computed and individual and local optimal values updated accordingly. Subsequently, crossover and mutation operations are applied to update particle speed and position based on the outcomes. Finally, algorithm convergence is assessed using predefined criteria. Simulation results indicate that compared to the traditional PSO algorithm, the proposed approach effectively mitigates local optima, significantly shortens overall path length, and reduces path planning time.

Furthermore, in [52], a path-tracking control strategy for unmanned surface ships is introduced, employing the recognition dynamics model. The particle swarm optimization technique is utilized to identify the linearized model using actual experimental data. Building upon this foundation, an adaptive control algorithm is proposed, incorporating waypoint navigation and speed directives. The efficacy of this approach is validated through experimental analysis conducted on a modified commercial fishing boat.

Additionally, Ref. [53] presents an emergency collision avoidance algorithm tailored for unmanned surface vehicles, utilizing a motion ability database. The objective function is tackled using a particle swarm optimization algorithm to determine the optimal initial velocity and rudder angle. The resulting solution corresponds to the emergency collision avoidance trajectory. Subsequently, collision avoidance parameters are computed based on the motion model described above. Experimental findings validate the efficacy of the proposed algorithm, suggesting its applicability for intelligent collision avoidance operations not only for unmanned underwater vehicles but also for other unmanned vessels.

To craft a top-tier multi-sensor integrated ASV path, Ref. [54] incorporates a combination strategy along with a traditional particle swarm optimization algorithm, introducing a greedy mechanism and 2-opt operation. The 2-opt optimization method involves iteratively swapping pairs of edges within a closed loop of points, aiming to minimize the total length of the loop. This process continues until further enhancement is no longer possible. At the outset, a deterministic black box is established for particle initialization, reducing the randomness inherent in traditional methods and eliminating many infeasible solutions. Greedy selection tactics and 2-opt operations are combined for local searches to uphold population diversity and avoid path redundancy. The improved algorithm is assessed against existing methods through Monte Carlo simulations across eight scenarios. Computational outcomes demonstrate that despite a slight sacrifice in computing efficiency, the improved algorithm exhibits superior performance, resulting in shorter routes and enhanced robustness.

Moreover, a novel three-layer local obstacle avoidance algorithm is introduced to address the issue of local obstacle avoidance [55]. Initially, real environment and obstacle models are established using polar coordinates. Subsequently, a known static path planning technique is integrated into particle swarm optimization (PSO). Next, the method is fused with navigation rules derived from the particle swarm optimization algorithm. Lastly, an obstacle avoidance approach is devised based on rolling windows within an unknown environment. A simulation experiment platform is then constructed to validate the feasibility and efficacy of these measures.

To enhance ship navigation path quality and augment the algorithm's global search capability, Ref. [56] introduces a hybrid particle–artificial bee colony algorithm. This approach not only improves the algorithm's global search prowess and accelerates convergence, but also enhances path quality and speeds up search rates.

The particle swarm algorithm utilizes particle movement to search for solutions, and there is no communication between particles. The implementation of this algorithm is typically complex, and more parameters are involved. Thus, it takes longer to plan a path with this algorithm. Also, there is a risk of encountering a local optimum in later stages.

Below are some differences between ant colony algorithm and swarm particle algorithm in the path planning of unmanned vessels shown in Table 2.

Table 2. Differences between ant colony algorithm and swarm particle algorithm in unmanned vessels' path planning.

Aspect	Ant Colony Algorithm	Particle Swarm Algorithm
Inspiration	Modeled after the foraging behavior of ants	Modeled after the social behavior of bird flocking
Solution Construction	Uses pheromone trails to build solutions	Utilizes the movement of particles to search for solutions
Communication	Communication through indirect pheromone sharing	No communication between particles
Exploration vs. Exploitation	Balanced exploration and exploitation based on pheromone levels	Exploration and exploitation guided by particle velocity
Complexity	Typically simpler implementation and fewer parameters	Typically more complex implementation and more parameters

3.2.3. Dynamic Path Generation-Based Algorithms

Dynamic path generation-based algorithms, a key tool in optimization and planning tasks, offer a distinct advantage over their static counterparts. Unlike static path planning algorithms, which precompute paths before execution and do not adapt to real-time changes, dynamic path generation algorithms continuously update paths based on evolving information. This dynamic approach allows for more flexible and adaptive planning, making them particularly useful in scenarios where conditions or requirements are subject to change. Examples of dynamic path generated-based algorithms include fast marching algorithm, RRT and RRT* algorithms, and D* Lite algorithm, etc.

1. Fast Marching Algorithm

The fast marching algorithm [57] is a highly efficient numerical method utilized for solving the Eikonal Equation (3). This equation, a nonlinear partial differential equation, is akin to an approximate wave equation. In the context of interface evolution problems, the resulting function T signifies the minimum time it takes for a curve to reach every point within the computational domain, given the velocity F $(\vec{x})$. By examining the set of T levels at various heights, one can consistently locate positions on the time evolution curve. Furthermore, when F = 1, the equation solution represents the distance field within the computational domain, referred to as the symbolic distance function:

$$\mathrm{F} \mid \nabla \mathrm{T} \mid = 1 \tag{3}$$

The fast marching algorithm employs a specific difference quotient instead of derivatives, as seen in Equation (4) for the two-dimensional scenario. Here, $\Delta + \mathrm{x}$ denotes a first-order forward difference operator for the independent variable x, and $\Delta - \mathrm{x}$ represents a first-order backward difference operator for x. While the concept behind the fast marching algorithm is akin to Dijkstra's algorithm, there's a key distinction: Dijkstra's algorithm updates nodes based on Euclidean distances, whereas the fast marching algorithm updates them using approximate partial differential equations derived from simplified functional equations.

$$|\nabla \mathrm{T}|^2 \approx \left(\max\{-\Delta_{+\mathrm{x}}\mathrm{T}, \Delta_{-\mathrm{x}}\mathrm{T}, 0\}\right)^2 + \left(\max\{-\Delta_{+\mathrm{y}}\mathrm{T}, \Delta_{-\mathrm{y}}\mathrm{T}, 0\}\right)^2 \tag{4}$$

Instead of adhering strictly to the interface, the fast marching method adopts a stationary approach to address the problem. To grasp this method, envision overlaying a grid onto the issue at hand, which is displayed in Figure 7. Suppose someone is stationed at every red grid point, each equipped with a watch. As the front passes each grid point, these individuals record the crossing time T. These recorded crossing times, forming a grid of values T(*x*,*y*), establish a function. At any point (*x*,*y*) on the grid, T(*x*,*y*) indicates the time when the front passed through that specific point.

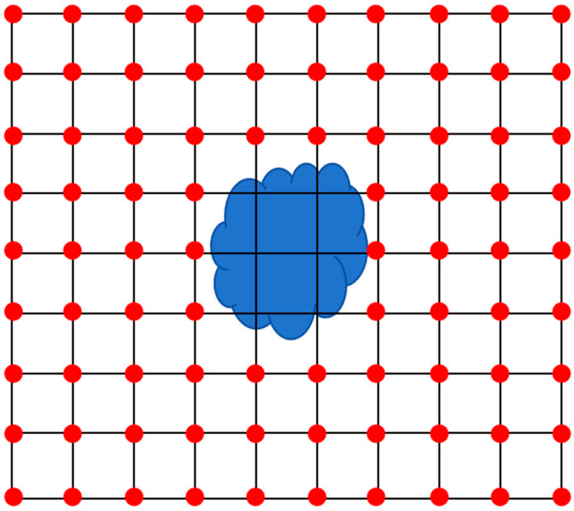


Figure 7. Schematic representation of fast marching algorithm.

Researchers present a novel computer-based algorithm designed to address the issue of USV formation path planning [58]. Utilizing the fast marching (FM) method as its foundation, this algorithm is specifically crafted to operate in dynamic environments, incorporating the innovative constrained FM method. By doing so, it accurately captures the dynamic movements of ships in motion while ensuring computational efficiency. This algorithm has demonstrated effective performance in navigating complex environments through evaluations conducted in a simulated area.

To solve the problem of under-actuated ASVs constrained by various motions, a new "Angle-guided fast marching square" algorithm is proposed in [59], which makes the generated path conform to the dynamics and direction constraints of water vehicles. The researchers also specifically studied the formation problem and developed an algorithm that enables USVs to form the desired shape from a trajectory of random initial configuration (position and orientation).

To mitigate the risk of potential collisions on the path, a prioritization scheme based on the nearest approach point distance (DCPA) was proposed and developed. To address the safety concerns and complexity associated with global path planning for surface unmanned ships, the fast marching method is improved [60]. To enhance the security of the planned route, obstacles must be installed. The time field function enables USVs to avoid obstacles at a distance. In order to simplify the planned route, the gradient inflection point evaluation function is set to minimize the number of turns for USVs.

The fast marching algorithm is based on wavefront propagation. It is complete and gains optimal solutions in a known environment. Moreover, it is moderate in implementation complexity and highly efficient in grid-based environments. Thus, path planning takes a comparatively short time. However, the issue is that the generated path is too proximate to obstacles, which means the security level is relatively low.

2. RRT and RRT* Algorithms

The RRT algorithm [61] serves as a method for generating open-loop trajectories for nonlinear systems with state constraints. Fast search random tree is an effective approach for exploring nonconvex, high-dimensional spaces through the random construction of space-filling trees. This method enables the swift and efficient navigation of high-dimensional spaces, guiding the search toward unexplored regions through random sampling points within the state space. It proves particularly useful for addressing path planning challenges in complex and dynamic environments for multi-degree of freedom robots. An example of RRT algorithm is displayed in Figure 8. In Figure 8a, the red point stands for the start point and the green point stands for the end point. On the premise that the start point and end point of the path are known, a sampling point x_{rand} is randomly generated, a node x_{near} that is closest to x_{near} is found on the tree, x_{rand} and x_{near} are connected, and node x_{new} is obtained on the line with step size E_i as the appearance tree node. In Figure 8b, collision detection during the RRT algorithm is presented. If a new branch, namely x_{near} and x_{new}, collides with an obstacle, the path is removed.

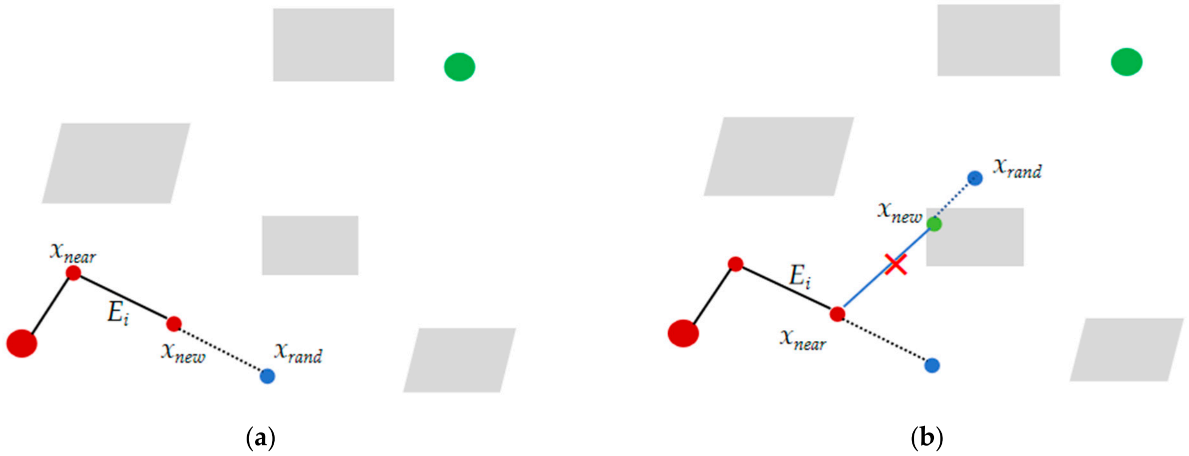
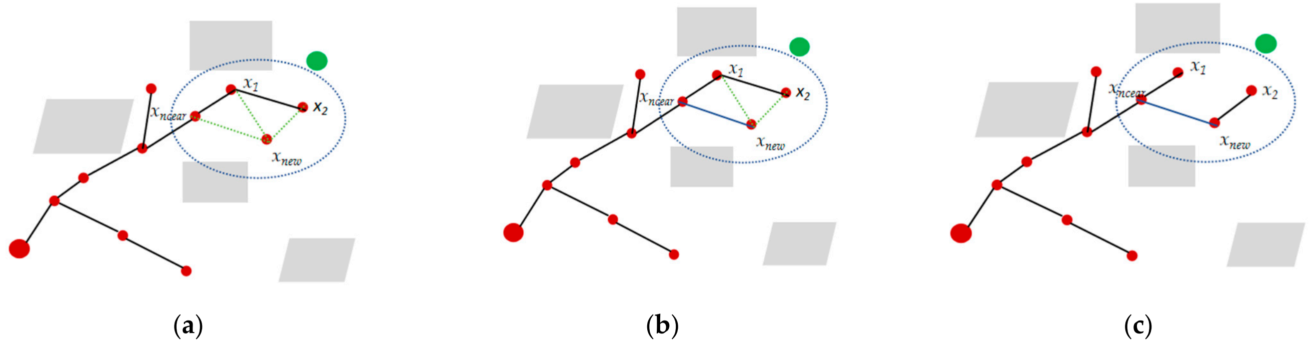


Figure 8. (**a**) Schematic representation of how RRT algorithm expands the tree; (**b**) collision detection during the RRT algorithm. The bigger red points stands for the start points and the bigger green points stand for the end points. The smaller red points are points that have already been traversed. The smaller green point is the new point which collide on obstacles. The blue points are the sampling points randomly generated.

RRT* algorithm is an asymptotic optimal algorithm, which is the optimization of the RRT algorithm. Asymptotically optimal means that as the number of iterations increases, the resulting path becomes increasingly optimized. The process of the RRT* algorithm is identical to that of the RRT algorithm, but there are two main differences: Firstly, the parent node for x_{new} is once again selected. Different from how the nearest is directly selected as the parent node of x_{new} in the RRT algorithm, RRT* selects the parent node for x_{new} again to minimize the starting cost of x_{new}. The selected node can be all the points connected near the node. Generally, the nearest node x_{near} is found within the defined radius r near the newly generated node x_{new} as the nearest replacement of x_{new}'s original parent node. The path cost from the starting point to each neighbor node x_{near}, as well as the path cost from the neighbor node x_{near} to x_{new}, must be calculated in sequence, and the neighbor node x_{min} with the lowest path cost is selected as the new parent of x_{new}. Secondly, once the parent node is reselected, all adjacent nodes of that node are rewired. The principle of routing is to minimize the cost of all nodes to the starting point. Figure 9 shows an example of how the RRT* algorithm works. In Figure 9a, the RRT* algorithm reselects the parent node here, draws a circle with x_{new} as the center of the circle and R as the radius, connects x_{new} with nodes x_{new}, x_1 and x_2 in the circle, and compares the way to reach x_{new} with the shortest path consumption. If the path cost to reach x_{new} through x_{near} is less than the path cost to reach X_1 through x_{near}, or if the path cost is less to reach x_1 and x_2 through x_{near} and then to reach x_{new}, then x_{near} is directly connected to x_{new}, which is displayed in Figure 9b. After the parent node is reselected, the rewiring operation is carried out. For surrounding nodes, the rewiring operation is to determine whether the distance consumption from the newly generated node to the surrounding node is smaller than that of the original path. For the path to the x_2 node in the figure, the distance consumption from the newly generated node x_{new} to x_2 is shorter than that of the original path. Then, the original path is rewritten, which is displayed in Figure 9c.



(**a**) (**b**) (**c**)

Figure 9. (**a**,**b**) Reselection of parent node in RRT* algorithm; (**c**) rewiring operation in RRT* algorithm.

In [62], the VF-RRT* algorithm is introduced to tackle path planning obstacles by incorporating virtual field sampling and ocean current constraint functions into the RRT* algorithm. Three sets of simulation experiments are conducted to evaluate its effectiveness. The results demonstrate that the algorithm produces more reasonable planning paths, highlighting its effectiveness in path planning and task allocation.

Also, the researchers of [63] utilize the RRT* and dynamic window approach (DWA) algorithms for path planning. The proposed method efficiently combines global and local planning to explore the water surface while ensuring USV safety. Simulation experiments confirm the effectiveness of the devised algorithm in effectively accomplishing water surface patrol missions under shallow water conditions.

A refined RRT-based formation path planning algorithm for unmanned surface vehicles (USVs) is proposed in [64], focusing on global path planning and obstacle avoidance. To maintain the shape stability of the USV formation, the algorithm introduces the concepts of “non-strictly conformal correction vector” and “non-strictly conformal control circle region” within the EXTEND operation of RRT. Additionally, the expansion direction of the search tree is biased toward the strictly conformal point. In case of sudden obstacles, the algorithm incorporates an adjustable collision avoidance circle and obstacle correction vector within the RRT collision inspection operation. This ensures that the USV formation can safely evade collisions to the maximum extent while preserving formation stability.

In [65], an enhanced RRT algorithm is presented for ship path planning within intricate inland waterways. This improved algorithm incorporates features for path shearing, sliding block, and maintaining a safe distance between the moving ship and obstacles. Moreover, the algorithm undergoes testing in two distinct inland waterway scenarios, confirming its feasibility and reliability. The fast search random tree (RRT) algorithm stands out as an efficient method for multi-dimensional space planning.

In [66], the enhanced RRT algorithm is employed for path planning tasks concerning unmanned ships, and its performance is compared against that of the traditional RRT algorithm. Experimental findings indicate that the improved RRT algorithm generates paths that are both shorter and smoother compared to those produced by the traditional RRT algorithm.

Aiming at the problems of blind expansion and low searching efficiency of traditional fast-expanding random tree algorithm, a path planning approach for unmanned ships is introduced in [67], combining the artificial potential field method with the fast-expanding random tree algorithm. The fast-expanding random tree algorithm can explore the path in any direction, and the improved sampling frequency is integrated with the new potential field function of the artificial potential field method so that the unmanned ship can move toward the target area and have the ability to stay away from obstacles.

Furthermore, Ref. [68] introduces a global path planning method for unmanned ships utilizing an enhanced BTO-RRT algorithm. This method aims to generate a smooth path with minimal energy consumption and a shortened path length in static environments. Initially, an adaptive step sampling strategy is proposed based on an artificial potential field to address issues encountered by the original BTO-RRT, such as an inadequate sampling of the fast search random tree and an inability to fit cubic splines. By incorporating adaptive step sizes, these challenges are effectively overcome, enabling the improved algorithm to produce optimized trajectories with reduced path lengths. The completeness of the generated path and the speed of path search are also taken into account. Ultimately, the optimized trajectory contributes to lowering the energy consumption of unmanned ships during autonomous navigation.

In [69], an enhanced RRT* algorithm is proposed to optimize path planning for unmanned ships, aiming to expedite planning and shorten sailing routes. Initially, the algorithm enlarges the target point, converting the single-point problem into a region-based one, and extends the target bias accordingly. Once an initial feasible path is computed, the sampling area is regenerated based on this path, enabling subsequent sampling within the redefined region. Upon obtaining the final feasible path, pruning and cubic B-spline interpolation are applied to refine and smooth the trajectory, aligning it more closely with the ship's intended route. Finally, simulation verification is conducted using PyCharm.

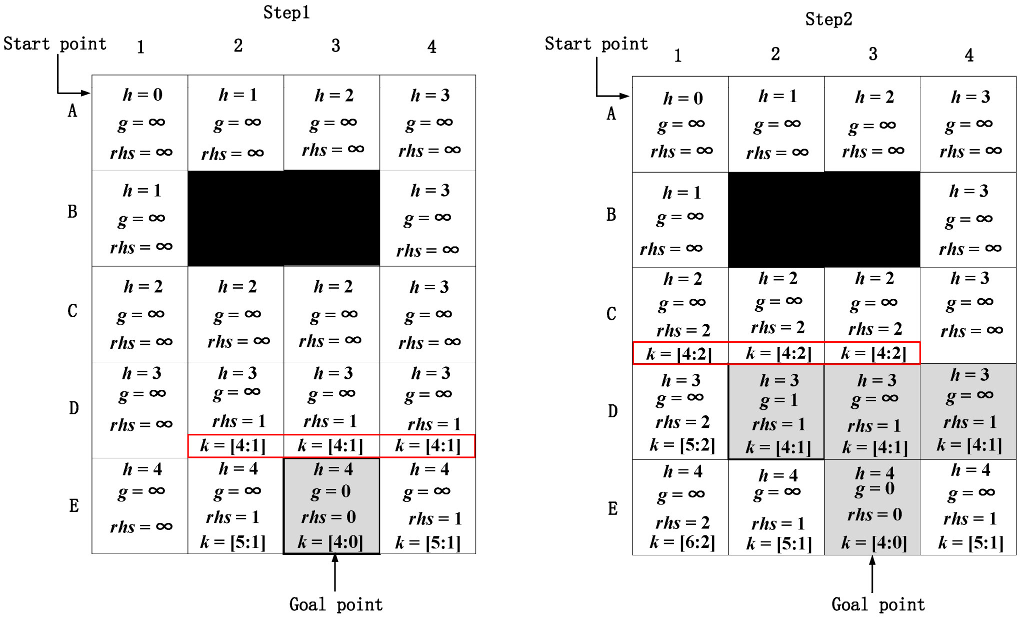
The RRT and RRT* algorithms are based on a random sampling and tree growth search strategy. They have a fast sampling speed and are suitable for unknown environments. Thus, the time of path planning is relatively short. However, the generated path is too long, which is a disadvantage of this algorithm.

1. D* Lite Algorithm

D* Lite [70] is a sophisticated path-finding algorithm with wide-ranging applications in robotics and game development. This algorithm efficiently tackles the shortest path problem and incorporates features like path replanning and correction. It operates on graph data structures comprising interconnected nodes. The leading node points to the router's current location, while the successor node points to the current node. D* Lite maintains two scores for each node in the graph. The G score represents the cost of reaching a particular point in the graph and denotes the shortest path from the start point to the current location. Initially, the G score for the start point is zero since the pathfinder is already there. Conversely, the RHS score serves as a heuristic estimate, predicting the next optimal location for the pathfinder. If the next location is obstructed, the connection cost is set to infinity, rendering the RHS score infinite as well. Leveraging these scores, the algorithm iteratively propagates a path from the end point to the start point, ultimately determining the optimal path from start to finish. Let S represent the finite set of nodes within a graph, and let Succ, a subset of S, consist of the nodes s, where s is an element of S. The path cost rhs(s) is the cost from the current node s to the goal node s_{goal}, calculated by Equation (5). Here, c(s, s') represents the cost of transitioning from node s to node s' within Succ(s); g(s') refers to the actual path cost from the extension node s' to the goal node s_goal. rhs(s) is updated prior to g(s), and the rhs(s) values for expanded nodes are revised as obstacles emerge or vanish. However, all nodes' g(s) values need not be updated concurrently with rhs(s).

$$\mathrm{rhs}(s) = \begin{cases} 0,\ if\ s = s_{goal} \\ \min\limits_{s' \in \mathrm{Succ(s)}} (\mathrm{c}(s, s') + \mathrm{g}(s')), \mathrm{otherwise} \end{cases} \tag{5}$$

Figure 10 illustrates the step-by-step process of the traditional D* Lite algorithm. In the diagram, the black grid represents obstacle nodes, while the light gray grid indicates the currently expanded node. When expanding a node from the goal node, the initial action involves initializing the node information. In Step 1, E3 is selected as the goal node for expanding its neighboring nodes. Three nodes with identical values of k1(s) are identified: D2, D3, and D4. All three nodes have the smallest k1(s) values, necessitating their progressive expansion. Moving to Step 2, four nodes require expansion, as shown by the gray grid. It is evident that at least four steps are needed to expand the node at layer D. After expanding the three nodes, three nodes with identical values of k1(s) emerge: C1, C2, and C3. These nodes are all the smallest among all the surrounding expanded nodes. Thus, the process repeats. If numerous nodes share the same values, more nodes must be expanded, leading to increased computation time and a decreased search efficiency of the algorithm.



Step1	1	2	3	4
A	$h=0$ $g=\infty$ $rhs=\infty$	$h=1$ $g=\infty$ $rhs=\infty$	$h=2$ $g=\infty$ $rhs=\infty$	$h=3$ $g=\infty$ $rhs=\infty$
B	$h=1$ $g=\infty$ $rhs=\infty$			$h=3$ $g=\infty$ $rhs=\infty$
C	$h=2$ $g=\infty$ $rhs=\infty$	$h=2$ $g=\infty$ $rhs=\infty$	$h=2$ $g=\infty$ $rhs=\infty$	$h=3$ $g=\infty$ $rhs=\infty$
D	$h=3$ $g=\infty$ $rhs=\infty$	$h=3$ $g=\infty$ $rhs=1$ $k=[4:1]$	$h=3$ $g=\infty$ $rhs=1$ $k=[4:1]$	$h=3$ $g=\infty$ $rhs=1$ $k=[4:1]$
E	$h=4$ $g=\infty$ $rhs=\infty$	$h=4$ $g=\infty$ $rhs=1$ $k=[5:1]$	$h=4$ $g=0$ $rhs=0$ $k=[4:0]$	$h=4$ $g=\infty$ $rhs=1$ $k=[5:1]$

Step2	1	2	3	4
A	$h=0$ $g=\infty$ $rhs=\infty$	$h=1$ $g=\infty$ $rhs=\infty$	$h=2$ $g=\infty$ $rhs=\infty$	$h=3$ $g=\infty$ $rhs=\infty$
B	$h=1$ $g=\infty$ $rhs=\infty$			$h=3$ $g=\infty$ $rhs=\infty$
C	$h=2$ $g=\infty$ $rhs=2$ $k=[4:2]$	$h=2$ $g=\infty$ $rhs=2$ $k=[4:2]$	$h=2$ $g=\infty$ $rhs=2$ $k=[4:2]$	$h=3$ $g=\infty$ $rhs=\infty$
D	$h=3$ $g=\infty$ $rhs=2$ $k=[5:2]$	$h=3$ $g=1$ $rhs=1$ $k=[4:1]$	$h=3$ $g=\infty$ $rhs=1$ $k=[4:1]$	$h=3$ $g=\infty$ $rhs=1$ $k=[4:1]$
E	$h=4$ $g=\infty$ $rhs=2$ $k=[6:2]$	$h=4$ $g=\infty$ $rhs=1$ $k=[5:1]$	$h=4$ $g=0$ $rhs=0$ $k=[4:0]$	$h=4$ $g=\infty$ $rhs=1$ $k=[5:1]$

Figure 10. The specific process of traditional D* Lite algorithm [71]. A-E and 1-4 in the figure stand for the coordinates of the grids. The red square shows at least four steps are needed to expand the node at layer D in step 1 and at least four steps are required to expand the node at layer C in step 2.

In [71], an upgraded iteration of the D* Lite algorithm was utilized for navigating paths between various target points within an unfamiliar obstacle-laden setting. By fine-tuning the heuristic function embedded within the D* Lite algorithm, efforts were made to minimize the expansion of nodes, consequently boosting search efficiency and refining the planned route. Through rigorous experimental validation, this algorithmic enhancement was juxtaposed against four alternative approaches across varying environments, ranging from straightforward to intricate. The outcomes of these experiments underscored the versatility and efficacy of the proposed methodology.

Research [72] introduces a self-navigating unmanned boat system tailored for wetland surveillance. Employing the D* algorithm alongside an enhanced line-of-sight (LOS) guidance law, the system orchestrates track control and global path planning for the unmanned vessel. Integration of the YOLOv3 algorithm aids in target recognition and data collection during navigation, facilitating autonomous obstacle evasion. Sensor data capture contributes to environmental monitoring, with information transmitted to a shoreline control center for automatic data acquisition and wetland surveillance.

The D* Lite algorithm relies on graph search and backtracking, offering an efficient approach to path planning. Notably, it demonstrates adaptability in dynamic environments, allowing for flexible adjustments. However, when the environment is complex, it would have many expanded nodes, which leads to reduced search efficiency, increased algorithmic time overhead, and a proliferation of inflection points. Thus, the time of path planning is comparatively long, and the distance traveled is relatively long, which also leads to higher energy consumption.

After comparing these algorithms, which belong to dynamic path generation-based algorithms, the differences between them are noted and listed in Table 3.

Table 3. Differences between fast marching algorithm, RRT and RRT* algorithms, and D* Lite algorithm in unmanned vessels' path planning.

Aspect	**Fast Marching Algorithm**	**RRT and RRT* Algorithms**	**D* Lite Algorithm**
Approach	Grid-based method	Sampling-based method	Graph-based method
Search Strategy	Wavefront propagation	Random sampling and tree growth	Graph search and backtracking
Completeness	Complete in a known environment	Probabilistically complete	Complete
Optimality	Optimal in a known environment	Probabilistically non-optimal	Optimal
Efficiency	Highly efficient in grid-based environments	Depends on sampling density	Efficient in graph-based environments
Usage	Suitable for known environments	Suitable for unknown environments	Suitable for dynamic environments
Implementation Complexity	Moderate	Moderate to high	Moderate

3.2.4. Machine Learning-Based Algorithms

Machine learning-based algorithms for unmanned vessels' path planning leverage computational models trained on data to determine optimal paths autonomously. These algorithms typically involve the following steps: data collection, feature engineering, model training, path prediction, path execution, etc.

1. Reinforcement Learning Algorithms

Reinforcement learning is when agents learn to navigate through trial and error, receiving rewards or penalties based on their actions and adjusting their behavior accordingly. Using reinforcement learning algorithms, the unmanned ship and the environment constantly exchange information to achieve self-learning.

In [73], a fresh approach to autonomous path planning for unmanned ships navigating unfamiliar surroundings is unveiled utilizing deep reinforcement learning methods. This model harnesses the deep deterministic policy gradient (DDPG) algorithm, empowering the agent to acquire optimal action plans through ongoing engagement with the environment and past experiences. Comparative testing highlights the model's accelerated convergence and steadfast stability in achieving autonomous path planning.

To achieve intelligent collision avoidance and path planning for maritime autonomous surface ships (MASSs) in unfamiliar surroundings, Ref. [74] introduces a reinforcement learning-based MASS path planning algorithm. Through simulation experiments conducted in the Rizhao Port area using Python (Https://www.python.org/ accessed on 20 March 2024) and the Pygame platforms, the method demonstrates adaptability, as evidenced by the results.

In [75], a path planning and maneuvering technique based on Q-learning is presented, enabling autonomous operation of cargo ships without human input. The Nonmoto model is employed to simulate the ship's behavior in waterways. Distances, obstacles, and exclusion zones are normalized to assess the ship's performance and maneuvering choices through rewards and penalties. Q-learning is then utilized to develop an action reward model, guiding the ship's movements based on learned results. By prioritizing higher reward values, the ship autonomously determines suitable paths and navigation strategies.

In [76], a coastal ship path optimization method is introduced, leveraging enhancements to the deep deterministic policy gradient (DDPG) and Douglas–Peucker (DP) algorithms. Initially, the DDPG network structure is refined with long short-term memory (LSTM) to utilize historical state data for precise action prediction in current environmental conditions. Additionally, upon addressing concerns regarding low learning efficiency and convergence rates associated with the traditional DDPG reward function, improvements have been made. Furthermore, to mitigate the excessive inflection points in the planning path and reduce navigation risks, an enhanced DP algorithm is proposed, refining the path planning for heightened safety and cost-effectiveness.

Additionally, Ref. [77] introduces a path planning and collision avoidance approach grounded in deep reinforcement learning (DRL) to address the path planning challenges encountered by unmanned vehicles amidst uncertain environments. Employing the deep Q-learning network (DQN), the system continually engages with a visual simulation environment to accumulate experiential data, enabling the agent to refine its optimal action strategies within this setting. To enhance the DRL algorithm, the artificial potential field (APF) algorithm is integrated to refine both the action space and reward function of the DQN algorithm. Simulation outcomes affirm the method's efficacy across varied conditions, demonstrating the enhanced DRL's capacity to autonomously execute collision avoidance path planning.

Furthermore, Ref. [78] introduces a path planning algorithm utilizing the deep deterministic policy gradient (DDPG). Through a succession of simulation experiments, it is demonstrated that the optimal path for unmanned surface vehicles (USVs) determined by the DDPG-based algorithm is not only faster, but also more precise compared to two alternative methods. These experiments underscore the notable time efficiency of the DDPG algorithm alongside its superior path quality and safety performance.

In [79], a marine search and rescue (SAR) environment model is constructed using field data and electronic charts. Subsequently, an autonomous coverage path planning model for marine SAR is introduced, employing reinforcement learning. Within this model, a multi-constraint reward function directs ship agent navigation behaviors. Throughout the iterative training of the path planning model, a nonlinear action selection strategy dynamically regulates the probability of random action selection, ensuring stable model convergence. Finally, experiments are conducted across various small-sea SAR simulation scenarios.

Furthermore, Ref. [80] introduces the marine search and rescue (SAR) vessel coverage path planning framework (SARCPPF) tailored for multiple persons in the water (PIWs). Comprising three key modules, namely drift trajectory prediction, multi-level search area environment modeling, and coverage search, this framework is designed to optimize SAR operations. Initially, a sea-scale drift trajectory prediction model is established utilizing random particle simulation. Subsequently, a hierarchical probabilistic environment diagram model is devised to facilitate SAR coordination among multiple units. Moreover, the integration of deep reinforcement learning with a multi-variable reward function guides ship agent navigation behaviors, enhancing operational efficiency. The developed overlay path planning algorithm aims to maximize success rates within constrained timeframes.

Reinforcement learning algorithms effectively address challenges presented by continuous state and action spaces. They operate by directly processing raw data inputs to generate actionable outputs, facilitating an end-to-end learning approach. This methodology significantly enhances algorithm efficiency and convergence, which means shorter path planning times.

2. Artificial Neural Network

An artificial neural network (ANN) [81] mirrors the architecture and functionality of the human brain in computational form. It comprises interconnected nodes, termed neurons or artificial neurons, arranged in layers. These neurons receive input signals, undergo processing via an activation function, and generate output signals, which can further propagate to other neurons. Additionally, several path planning techniques draw inspiration from ANN principles.

In [82], unmanned ships are conceptualized as particles, with an artificial neural network (ANN) employed to compute collision penalty functions for distribution points and obstacles. To determine the optimal path, an evaluation function incorporating path length and collision penalties is devised, seeking to minimize its value. Furthermore, the simulated annealing technique is introduced to enhance the activation function of the neural network's output layer, thereby improving classification accuracy and mitigating issues related to local minima.

The neural network algorithm is a prominent intelligent approach known for its precise data processing capabilities. Leveraging neural networks' swift parallel computing prowess is particularly apt, especially in addressing local extremum challenges encountered in path planning with multiple obstacles using the artificial potential field method.

3.3. Path Planning Algorithms Taking into Account Characteristics of Sailboats

Compared to the path planning algorithms for unmanned vessels, path planning algorithms for unmanned sailboats have the following differences.

1. Wind Consideration

Unmanned sailboats heavily rely on wind power for propulsion. Thus, path planning algorithms for sailboats need to incorporate wind direction and strength into their calculations. Unlike traditional vessels, which might prioritize direct routes or fuel efficiency, sailboats take more circuitous routes to take advantage of favorable wind conditions. An example of a speed polar diagram, which indicates the highest sustainable velocity of the vehicle under specific wind speed and direction conditions is displayed in Figure 11. The upwind no-go zone, covering angles from −45 to 45 degrees, is disregarded. Similarly, the downwind no-go zone is highlighted, spanning from 150 to 210 degrees. Although the boat technically can sail in these areas, they are avoided in practice due to instability-related safety concerns. These speed polar diagrams and exclusion zones are essential references for the local path planning algorithm [83].

2. Tacking and Gybing

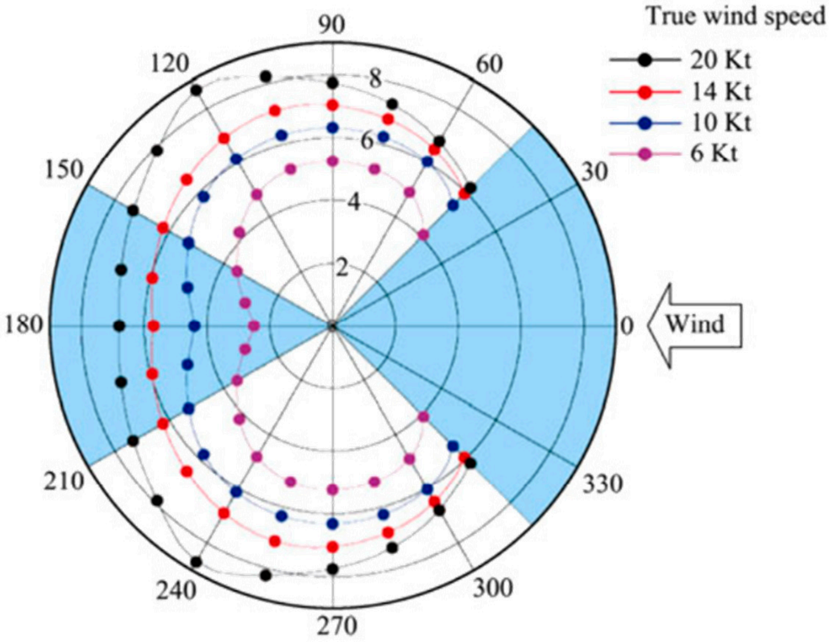


Figure 11. A speed polar diagram illustrating the maximum velocity a vehicle can achieve under steady-state conditions for different combinations of wind speed and wind direction [4].

Tacking and gybing are essential sailing maneuvers. Tacking involves sailing against the wind in a zigzag pattern while gybing changes direction when the wind is behind the boat. Path planning algorithms for sailboats need to include these maneuvers to optimize navigation. By adjusting the route based on wind direction and speed, sailboats can efficiently harness wind power to reach their destination. Integrating tacking and gybing ensures the most effective use of wind energy, allowing sailboats to navigate efficiently and safely. For instance, a hysteresis potential is added in [84] to consider the impact of tacking and gybing maneuvers. This accounts for the additional cost associated with changing direction while sailing. Integrating this factor ensures that path planning algorithms optimize routes while minimizing the need for frequent tacking and gybing, enhancing overall efficiency and performance.

3. Energy Management

Path planning algorithms for unmanned sailboats must efficiently manage energy resources due to their reliance on intermittent wind energy. This involves considering factors such as battery charge levels, availability of sunlight for solar panels (if equipped), and the energy demands of propulsion and onboard systems. By optimizing energy usage and considering environmental conditions, these algorithms ensure safe and autonomous navigation to the sailboat's destination, even in challenging conditions.

The differences in path planning algorithms for unmanned vessels and sailboats are summarized in Table 4.

Table 4. Differences between path planning algorithms for unmanned vessels and sailboats.

Aspect	Unmanned Vessels	Unmanned Sailboats
Wind Consideration	Consider wind, but less emphasis	Wind direction and strength crucial for route planning
Tacking and Gybing	Linear navigation; minimal tacking or gybing	Essential maneuvers; path planning optimizes wind use
Energy Management	Continuous power (e.g., fuel)	Rely on intermittent wind; manage energy efficiently

Researchers have also produced multiple path planning algorithms for unmanned sailboats in recent years. These algorithms can be categorized as graph-search-based methods, randomization methods based on optimization techniques, potential field-based methods, and machine learning-based methods.

3.3.1. Graph-Search-Based Methods

Graph-search-based algorithms serve as the path planning method for unmanned sailboats, employing a graph for navigation. In this spatial discretization, nodes represent the sailboat's origin, destination, and reachable locations, while edges denote possible travel paths between nodes. Examples of these algorithms include the A* algorithm and the Dijkstra algorithm.

1. A* Algorithm

Just as the A* algorithm is applicable in path planning for unmanned vessels, it can also serve in the path planning for unmanned sailboats. In [75], every node is linked to its eight closest neighbors, and the wind vectors are depicted as bold arrows in Figure 12. Nodes located on land masses or where the wind speed exceeds the limit are excluded. Routing is performed using an A* algorithm, incorporating a heuristic for performance enhancement. Specifically, the optimal node xi is determined not solely based on the lowest cost c(x_i), but also considering the lowest value of c(x_i) + h(x_i). Here, h(x_i) represents the distance from node x_i to the destination node x_{dest}, divided by the boat's hull speed, which algorithm users can adjust.

In [85], an autonomous obstacle avoidance system is devised for unmanned sailboats, tailored to meet obstacle evasion needs. The system introduces an obstacle avoidance path planning algorithm rooted in A* and a tracking control algorithm based on line-of-sight (LOS).

The simulation test and sailing test of the case sailboat successfully realized autonomous obstacle-avoiding sailing under multiple obstacle scenarios, which fully proved the feasibility and real-time performance of the method.

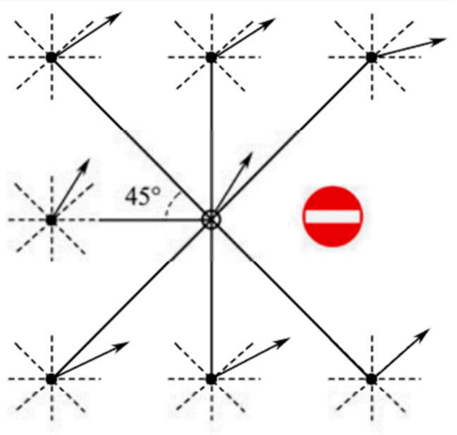


Figure 12. A part of the routing graph. Edges to and from the center node are denoted as solid lines, while edges connecting the surrounding nodes are denoted as dashed lines [4].

Researchers in [86] studied three perception, planning, and control modules, respectively. At the planning level, considering obstacle information, sailing characteristics, and wind field information, an obstacle avoidance algorithm for unmanned sailboats based on A* is proposed, which can plan the local path of sailing ships in real time. The obstacle avoidance path planning algorithm based on A* can quantify the loss of steering and wind direction and plan a new route according to real-time conditions. The rationality of the route has been proven by simulation and real ship tests.

In [87], a path planning strategy employing the A* algorithm is crafted to align with the speed pole curve of unmanned sailboats, facilitating an independent exploration of optimal routes. The motion control algorithm based on the Mamdani fuzzy inference machine is combined to realize the goal of short-distance unmanned navigation. Experiments verify the path planning algorithm and realize the goal of short-distance unmanned navigation.

In [88], the algorithm suggested is rooted in the A* algorithm and dynamically adapts the routing diagram to accommodate fluctuating weather conditions. The algorithm is executed using the declarative rule-based programming language Constraint Processing Rules (CHR). The implementation significantly reduces the computation time compared to existing commercial applications. The system utilizes actual wind and current forecasts, takes into account diverse parameters of the sailboats, and offers a graphical user interface.

The A* algorithm serves as a direct search approach for determining the shortest path within a static road network and stands out as the premier heuristic algorithm in this regard. As long as the estimated heuristic loss is not higher than the actual loss, the route with the minimum total loss between the initial node and the destination node can be found through search. Therefore, it consumes less energy when traveling along the path planned by using this algorithm.

2. Dijkstra Algorithm

The Dijkstra algorithm has also been adopted for path planning for unmanned sailboats. In [89], the Dijkstra algorithm is employed to calculate the globally optimal path within the routing graph. Unlike the A* algorithm, Dijkstra does not necessitate the inclusion of heuristic elements to identify the node with the lowest cost. However, two extra constraints are integrated: initially, edges intersecting or touching the coastline are omitted from the graph, and the specific draught of the sail class is meticulously taken into account to avoid grounding.

In [90], the high-level control employs a customized cost function and the PRM-Dijkstra algorithm to navigate the unmanned sailboat (global path planning). Leveraging the sailboats' movement and wind distribution on the map, the algorithm optimizes route planning. Path planning algorithms like RRT and Dijkstra utilize this cost function to conduct global path planning for sailboats, considering their movement and the wind distribution on the map. These adjustments enable path planning methods like Dijkstra to efficiently find less time-consuming paths between initial and final locations, generating waypoints for local path planning.

The conventional Dijkstra algorithm relies on a breadth-first search without a heuristic function, resulting in lower efficiency due to exhaustive exploration. Hence, the time of path planning in this algorithm is relatively long. Additionally, it often produces unnecessary inflection points. As a result, the distance traveled is longer, and energy consumption is higher.

3.3.2. Randomization Method Based on Optimization Technique

Optimization-based randomization techniques are also applied in unmanned sailboat path planning. Among them, the most commonly used algorithm is ant colony optimization.

In [91], an improved ant colony optimization (IACO) technique is presented, specifically designed to suit the movement dynamics of unmanned sailboats. This method enhances the heuristic information function by considering parameters like path length, wind field height, and turning frequency alongside optional constraints. Additionally, a revised pheromone update rule incentivizes the ant colony to favor global paths characterized by shorter lengths and lower wind field height standard deviations. Simulation outcomes validate the efficacy of the optimized IACO algorithm in determining the optimal global path for unmanned sailboats, showcasing enhanced convergence.

Furthermore, in [92], the researchers introduce an enhanced adaptive ant colony algorithm tailored for path planning of unmanned sailboats. The objective is to achieve precise path planning for sailboats navigating through complex environments, considering their nuanced motion patterns and a distinct current path evaluation index. The effectiveness and superiority of the proposed unmanned sailing route are subsequently verified. Additionally, an improved ant colony optimization (ACO) approach is suggested for the global path planning of unmanned sailboats. This algorithm integrates enhanced environment modeling, taking into account wind direction, and aligns with sailboat motion characteristics by imposing constrained turning conditions.

Additionally, enhancements to the original heuristic function of ACO include factors such as path length, wind direction angle, and number of turns aimed at optimizing the sailboat's path for better length and consistent wind direction [93]. Furthermore, updates to the pheromone rules are proposed. Simulation comparisons with traditional ACO demonstrate the effectiveness of the improved ACO algorithm in obtaining an optimal global path for unmanned sailboats, showcasing improved convergence.

Furthermore, in [84], the researchers introduce a path planning method for unmanned sailboats employing the fusion algorithm of enhanced adaptive ant colony optimization and the rolling window method (RIACO). With the goal of achieving globally optimal paths while enabling real-time obstacle avoidance, the ant colony algorithm is enhanced to optimize path length, duration, and safety performance. Additionally, Bezier curves are utilized to smooth path inflection points, enhancing the navigability of unmanned sailboats.

To address challenges such as low convergence rates at the initial stage, susceptibility to local optimal solutions, and slow solution speeds, extensive experimental data analysis is conducted to assign key parameters α, β, M, and ρ in the ant colony algorithm. Subsequently, [94] presents an enhanced ant colony algorithm incorporating dynamic parameters and a novel pheromone-updating mechanism.

The ant colony algorithm constructs solutions using pheromone trails and communicates through indirect pheromone sharing. It boasts a more straightforward implementation and involves fewer parameters. Nonetheless, drawbacks like local optimization and sluggish convergence speed are encountered. Therefore, the time required to plan the path is comparatively long.

3.3.3. Potential Field-Based Methods

Different from path planning methods for unmanned vessels, there is a special kind of method for unmanned sailboats: the potential field-based method [95]. Potential field-based methods for unmanned sailboats are path planning algorithms that rely on the concept of potential fields to guide the motion of the sailboat. In these methods, the sailboat is treated as a point in a field where each point in the surrounding environment exerts a "force" on the boat, either attracting or repelling it. The potential field typically comprises two components: attractive and repulsive forces. Attractive forces guide the sailboat toward its goal or target destination, while repulsive forces help it to avoid obstacles or hazards. By summing up these forces, the sailboat can navigate through its environment while avoiding collisions and reaching its intended destination. Consequently, a favorable potential $\mathrm{U_g}$ is linked with the goal (the current waypoint), determined by Equation (6), where $\mathrm{G_g}$ denotes a constant weighting factor, and $\mathrm{d_g}$ represents the Euclidean distance from the boat's position P to the current waypoint position $\mathrm{P_g}$.

$$\mathrm{U_g} = \mathrm{G_g d_g} \tag{6}$$

Also, a repulsive potential $U_o^{(j)}$ is associated with every obstacle j and is defined as shown in Equation (7). Here, $\mathrm{G_0}$ is a constant weighting factor, $d_0^{(j)}$ is the Euclidean distance between the boat position P and the obstacle position $P_0^{(j)}$, and $\mathrm{d_{inf}}$ is the distance of influence of the obstacle. The potential approaches infinity when the boat is in close proximity to the obstacle to ensure collision avoidance. The onboard navigation sensors periodically update the list of detected obstacles.

$$U_o^{(j)} = \begin{cases} \mathrm{G_0}\left(\frac{1}{d_0^{(j)}} - \frac{1}{d_{inf}}\right), & if \ d_0^{(j)} \leq d_{inf} \\ 0, & otherwise \end{cases} \tag{7}$$

In [96], the researchers introduce a novel reaction path planning approach that accounts for designated no-go zones by converting them into virtual obstacles. Subsequently, the potential field algorithm is utilized to guide the sailboat toward the target while circumventing obstacles, particularly when the target lies directly upwind or downwind.

Additionally, in [97], the artificial potential field method is employed for the local path planning of unmanned sailboats. The environment and specific sailing constraints, such as upwind and downwind no-go zones, are represented by localized potentials centered around the vessel's position. These potentials are regularly updated to account for variations in wind direction and obstacle positions. A line-of-sight (LOS)-based potential ensures that the vehicle remains within a band surrounding the direct route between consecutive waypoints. To achieve a smoother trajectory while avoiding obstacles, a repulsive potential incorporating a velocity-dependent component is proposed.

Furthermore, leveraging the potential field method, the path planning algorithm creates a localized potential around the sailboat's location, considering both upwind and downwind constraints. This algorithm recalculates the course periodically to adapt in real-time to fluctuating wind conditions and identified obstacles [98].

In [83], a novel potential field method is introduced for navigating and controlling unmanned sailboats, designed to accommodate the unique kinematic constraints of sailboats. This method incorporates virtual obstacles surrounding the boat to depict upwind and downwind no-go zones, effectively translating specific sailing limitations into virtual barriers. Additionally, hysteresis is integrated to assess the costs associated with heading maneuvers (turning the bow to the wind) and to turn maneuvers (turning the stern to the wind), enabling the calculation of the optimal heading while ensuring safety relative to obstacles. Field test experiments were conducted to validate the various components of the system, marking an initial stride toward achieving fully autonomous and operational navigation for sailboats.

In [99], recognizing the uncontrollable speed of sailing ships, collision avoidance behavior is reframed as a course control issue. The gradient descent direction of the total potential field is utilized as the anticipated collision avoidance course. In situations where the sailboat faces no collision risks or sails downwind, only the time potential field is constructed to steer clear of stationary obstacles and reach the target point. For sailboats encountering navigational constraints, Gaussian likelihood functions are devised for specific encounter scenarios based on the International Regulations for Collision Avoidance at Sea (COLREGS) to dynamically construct obstacle potential fields. Simulation results demonstrate the efficacy of the proposed algorithm in executing collision avoidance maneuvers with other sailing ships and constrained motor vessels across diverse encounter scenarios, adhering to collision avoidance regulations and evading sailing dead zones.

In order to make use of the seasonal variation rules of ocean wind sites to reduce sailing time in path planning under variable wind fields, a probability potential field (PPF) was proposed in [100], combining the regular change in wind direction with the intensity of its influence on sailing movement to establish a spatial wind direction probability potential field. The probability potential field is taken as a component factor of the heuristic function to ensure that the algorithm tends to choose the solution that makes the wind direction probability potential field larger. Simulation results indicate that, in contrast to conventional real-time path planning algorithms, the real-time path planning algorithm leveraging the probability potential field enhances the average sailing speed and notably diminishes the time necessary for an unmanned sailboat to reach its destination.

In [101], an improved fast random search tree (RRT) method based on a probabilistic potential field is proposed according to the velocity pole curve of the sail, aiming at the sailing constraint and speed limit of the sail and the difficulty of obtaining the path in the uneven wind field diagram. It not only has the characteristics of an RRT random search but can also explore the path according to the speed characteristics of the sailboat based on probability, and quickly achieve the effective sub-optimal sailing route of the sailing boat in a non-uniform wind field environment. The average calculation results of many actual wind field planning experiments show that the proposed path planning method based on probability potential field has faster calculation speeds and a shorter planned sailing time than the conventional A* method and other improved RRT methods.

The potential field method is a popular choice for path planning in unmanned sailboats. This technique involves a robot's movement, depicted as a particle, being influenced by a field containing two main elements: an attractive potential guiding the robot toward the goal and a repulsive potential pushing it away from obstacles. Although effective, potential field methods can sometimes fall prey to local minima. Yet, in marine environments, especially during ocean monitoring tasks, instances of sparsity are common, which mitigates the likelihood of encountering local minima.

3.3.4. Machine Learning-Based Algorithms

Some algorithms based on machine learning are also used in the path planning of unmanned sailboats. The most commonly used are reinforcement learning algorithms.

The reinforcement learning method, specifically Q-learning, operates on a reward matrix and a set of actions that account for shifts in wind direction, allowing for an interpretation of the dead zone—where the sailboat struggles to gain speed against the wind. In [102], the proposed algorithm generates both straight and zigzag paths based on wind direction. These paths ensure the sailboat's safety and resilience, enabling prolonged sailing with reliance solely on predefined starting and destination points. The research demonstrates the effectiveness of this reinforcement learning algorithm by considering the sailboat's kinematics, producing feasible paths while circumventing mapped obstacles. The strategy has been successfully validated in simulation environments.

Reinforcement learning algorithms successfully tackle the complexities of continuous state and action spaces by processing unprocessed data inputs to produce practical outcomes, supporting a comprehensive end-to-end learning strategy. Thus, the time of path planning for this algorithm is comparatively short.

3.4. Path Planning Algorithms Taking into Account Marine Environmental Factors

In real applications, path planning algorithms for ASVs need to consider marine environmental factors such as ocean currents, wind power, waves, tides, infrastructure, weather conditions, etc. These factors may affect the results of the path planning algorithms mentioned above. It is of great significance to consider these influencing factors to ensure safety and effectiveness during path planning.

Research [103] describes a novel trajectory planning technique for autonomous surface vehicles (ASVs). This method focuses on optimizing energy consumption while navigating through environments containing arbitrary polygonal obstacles. Initially, the approach generates convex regions represented as halfspace descriptions based on an initial guess. This process effectively transforms nonconvex obstacle configurations into a smoother and convex format. The optimization problem operates within a continuous state space and is designed to accommodate external disturbances such as wind or ocean currents. Through comparative simulations against two similar methodologies, the proposed approach demonstrates significant enhancements in computational efficiency, thus validating its effectiveness.

Research [104] presents an enhanced version of the artificial fish swarm algorithm (IAFSA), focusing on four key modifications: (1) The introduction of a directional operator to enhance efficiency. (2) The proposal of a probability weight factor to adjust random behavior frequency, mitigating local optima. (3) The application of an adaptive operator for improved convergence performance. And (4) the utilization of the waypoint modifying path smoother to enhance path quality. The integration of IAFSA into the guidance, navigation, and control (GNC) system of a model ship is followed by a computer-based sea trial in the Nan Hai area. Environmental disturbances such as wind, waves, and currents are considered. Results indicate the suitability of IAFSA for practical applications.

In study [105], the C-Enduro unmanned surface vehicle (USV) is designed for extended maritime missions, with durations of up to 3 months. To enhance its endurance, an energy-efficient path planning algorithm has been developed. This algorithm integrates the Voronoi diagram, Visibility algorithm, and Dijkstra search algorithm, along with the inclusion of sea current data. By utilizing these elements, the algorithm calculates a collision-free and energy-efficient path, thereby enhancing the safety and operational capability of the USV.

Ocean currents and wind conditions significantly influence the energy consumption of UUVs, which have become the vital path planning methods that affect a UUV's navigation performance and endurance. A novel energy-efficient path planning algorithm, presented in [106], integrates artificial intelligence techniques while adhering to the Convention on International Regulations for Preventing Collisions at Sea (COLREG) guidelines. This algorithm accounts for energy consumption by incorporating wind and ocean current data to determine the optimal route.

Addressing the limitations of the conventional A-Star algorithm in ship path planning, an enhanced version is introduced in [107]. Initially, a comprehensive obstacle risk model, encompassing factors like water flow, lane separation, berthing, shore obstacles, and vessel mobility constraints, is formulated. Subsequently, the improved A-Star algorithm is delineated, detailing the generation of regular paths and berthing paths separately. Furthermore, a solution to seamlessly integrate these paths is provided.

It can be found that in the algorithms mentioned above, the environmental conditions such as wind and wave can be planned in advance, but they might change drastically in real-time. Currently, the real-time path planning is still a difficulty of research in this field, but there are some potential solutions that may be used to solve this question as a reference.

1. Senor Fusion

Combine data from multiple sensors, such as GPS, inertial measurement units (IMUs), and environmental sensors (e.g., anemometers for wind speed measurement, wave sensors), to accurately estimate the current state of the environment and adjust the planned path accordingly. Presently, open-source mapping algorithms primarily depend on two sensor types: LiDAR sensors and vision sensors [108]. Furthermore, integrating sensor fusion is becoming a foundational approach for constructing robust systems with various sensor combinations, enhancing the capabilities of mobile robots [109].

For instance, to fulfill the real-time and precise demands of multi-robot path planning in dynamic environments, research [110] employs a wireless sensor network for robot and obstacle localization. Additionally, an enhanced artificial intelligence algorithm is utilized for path planning.

What is more, research [111] introduces a robust approach for localizing objects in front of an autonomous underwater vehicle in three-dimensional space. This method leverages camera imagery, sonar imagery, and odometry data obtained from onboard sensors. The localization process involves employing diverse image processing techniques, along with a hybrid sonar/camera particle filter for calibration and fusion steps.

Also, in research [112], various sensor signals are efficiently fused to ensure robust anomaly detection, even in field environments where sensor occlusion may occur. The experiments with field robot data show better performance in identifying failures compared to previous methods. The proposed model can detect anomalous behaviors in real-time while also maintaining a low false detection rate, especially in cluttered fields.

In research [113], a straightforward yet efficient deep reinforcement learning (DRL) approach featuring our self-state-attention unit is introduced. A solution that enables the navigation of a tall mobile robot, measuring one meter in height, using low-cost devices such as a 2D LiDAR sensor and a monocular camera is proposed.

2. Dynamic Path Planning Method

Some dynamic path planning methods which have been used in land robots may also be used in the path planning of ASVs by utilizing algorithms that continuously update the planned path based on real-time sensor data. These algorithms should consider the current state of the environment, including wind speed, wave height, and direction, to calculate an optimal path.

For example, the rolling window planning approach, detailed in [114], encompasses environmental information prediction, local rolling optimization, and feedback information correction. Dynamic path planning for unmanned sailboats based on rolling windows involves real-time planning, leveraging local environmental data obtained through forward rolling detection. Within each planning window, the sailboat determines an optimized subgoal using heuristic methods derived from locally detected obstacle information. Subsequently, it performs local planning within the current window before transitioning to the next window in a rolling fashion to continually gather updated environmental data.

Also, to address dynamically unforeseen environments, a local hierarchy is formed using fuzzy decision-making (FDM) and fine dynamic window (FDW) layers in [115]. These layers manage large- and close-range collision avoidance by controlling surge and yaw velocity guidance signals, respectively.

In [116], considering the robustness and strong search capabilities of ant colony optimization along with the superior planning effectiveness of the rolling window method (RWM) in local path planning, a fusion algorithm called RACO is introduced. This algorithm aims to efficiently and safely navigate through complex dynamic environments to reach designated target areas promptly.

In addition, research [117] introduces an enhanced version of the dynamic window approach (DWA) utilizing Q-learning. Initially, we enhance the original evaluation functions by introducing two additional ones to improve global navigation performance. Subsequently, we define the state space, action space, and reward function for the Q-learning algorithm adopted in robot motion planning, striking a balance between effectiveness and speed. Following this, we adaptively learn the parameters of the proposed DWA using Q-learning, resulting in a trained agent capable of adapting to unknown environments.

3. Model Predictive Control (MPC)

MPC is a control strategy that utilizes a dynamic model of the vehicle and its environment to predict future states and optimize control inputs. By continuously updating the model with real-time sensor data, MPC can adjust the planned path to account for changing conditions.

In research [118], a unified path planning approach utilizing MPC to automatically determine maneuver modes is proposed. Safety is ensured by modeling surrounding vehicles as polygons and incorporating constraints in MPC to enforce collision avoidance between the ego vehicle and its surroundings. Additionally, to achieve smooth and natural maneuvers, a lane-associated potential field is integrated into the MPC's objective function. This method can be used for reference in the path planning of ASVs.

Research [119] presents an obstacle avoidance controller for autonomous vehicle navigation, employing nonlinear model predictive control. Initially, a reference trajectory is predefined using a sigmoid function, adapting to varying road conditions. When unexpected obstacles arise along this trajectory, dynamic adjustments are made to the reference trajectory. To handle dynamic obstacles, a moving trend function predicts changes in obstacle positions within the predictive horizon. Additionally, a risk index is incorporated into the cost function to enable collision avoidance, taking into account the relative positions of vehicles and obstacles within the predictive horizon.

To enhance path quality and evade unpredictable moving obstacles, a nonlinear model predictive control is employed to optimize the path [120]. Additionally, a secondary path planning process is conducted, taking into account lateral stability considerations.

Research [121] introduces a model predictive controller, termed NNSEM-MPC, which utilizes neural network simulation error minimization to predict the dynamics of autonomous surface vehicles (ASVs) accurately, particularly in the presence of wind disturbances.

4. Combination of Methods Mentioned Above

In some research, the methods mentioned above are combined to deal with the real-time path planning problem. For example, a hybrid algorithm, combining an enhanced A* algorithm with the dynamic window approach, is introduced for the real-time path planning of mobile robots in greenhouses in research [122]. Also, utilizing ultrasonic sensors, local obstacle avoidance is achieved to enable real-time optimal path planning.

4. Discussion

This review summarizes path planning methods for both unmanned vessels and unmanned sailboats. The major difference lies in the susceptibility of unmanned sailboats to external factors, making their navigation more challenging. Conversely, unmanned vessels rely solely on wind energy, enabling them to undertake prolonged tasks in remote areas without easy access to supplies. Thus, unmanned vessels find broader applications in civilian domains. It can be found that both categories utilize some shared methods but also employ distinct approaches in path planning. Using the indicators mentioned in Section 2.3, each algorithm is evaluated qualitatively, and the result is displayed in Tables 5 and 6.

Table 5. Evaluation of different path planning algorithms for unmanned vessels using four indicators mentioned in Section 2.3. Note: "+" indicates a high degree, "–"indicates a low degree, and "○" means ordinary.

Path Planning Algorithms for Unmanned Vessels					
Algorithms		**Time of Path Planning**	**Distance Traveled**	**Energy Consumption**	**Security Level**
Graph-Search-Based Algorithms	1. Dijkstra Algorithm	+	+	+	+
	2. A* Algorithm	–	–	–	+
Randomization Method Based on Optimization Technique	1. Ant Colony Algorithm	+	○	○	+
	2. Particle Swarm Algorithm	+	○	○	+
Dynamic Path Generation-Based Algorithms	1. Fast Marching Algorithm	–	–	○	–
	2. RRT and RRT* Algorithms	–	+	+	+
	3. D* Lite Algorithm	+	+	+	+
Machine Learning-Based Algorithms	1. Reinforcement Learning Algorithms	–	○	○	+
	2. Artificial Neural Network	–	○	○	+

Table 6. Evaluation of different path planning algorithms for unmanned Sailboats using four indicators mentioned in Section 2.3. Note: "+" indicates a high degree, "–"indicates a low degree, and "○" means ordinary.

Path Planning Algorithms for Unmanned Sailboats					
Algorithms		**Time of Path Planning**	**Distance Traveled**	**Energy Consumption**	**Security Level**
Graph-Search-Based Algorithms	1. A* Algorithm	–	–	–	+
	2. Dijkstra Algorithm	+	+	+	+
Randomization Method Based on Optimization Technique	Ant Colony Optimization	+	○	○	+
Potential Field-Based Methods	Potential Field-Based Methods	–	○	○	+
Machine Learning-Based Algorithms	Reinforcement Learning	–	○	○	+

Table 5 outlines the assessment of various path planning algorithms designed for unmanned vessels. Regarding the time of path planning, the A* algorithm, fast marching, RRT and RRT* algorithms, and machine learning-based algorithms demonstrate great advantages. As for distance traveled and energy consumption, the A* algorithm has outstanding performance among these algorithms. From a security level point, all algorithms perform well except for the fast marching algorithm. Also, the evaluation of different path planning algorithms for unmanned sailboats is summarized in Table 6. It can be observed that the A* algorithm, potential field-based methods, and reinforcement learning perform better in terms of time of path planning. Similar to algorithms for unmanned vessels, the A* algorithm outweighs other algorithms in terms of distance traveled and energy consumption. Moreover, all algorithms have a high-security level.

Since different ASVs have different ranges, cruise speeds, maneuvering speeds, payloads, limitations of propulsion, and weight, a comparison based on these specific criteria is also discussed. In terms of range, algorithms like A* and Dijkstra's algorithm are suitable for long-range path planning as they focus on finding the shortest path between two points. They are efficient for planning paths across large distances without excessive computational burden. Algorithms like RRT and RRT* are more suitable for short-range path planning, especially in complex and dynamic environments. They excel in exploring the nearby space rapidly and finding feasible paths in cluttered areas [7,8]. As for cruise speeds, there's no direct correlation between path planning algorithms and cruise speeds. Similar to cruise speeds, path planning algorithms do not inherently dictate maneuvering speeds. Also, path planning algorithms are agnostic to payloads and they typically do not directly consider the limitations of propulsion or weight.

It can be found that different types of heuristics inspired by nature have been used, such as the ant colony algorithm, the particle swarm algorithm, etc. Most of them are based on biological processes. In recent years, a multitude of emerging water-based metaheuristics have been successfully used in path planning problems [123]. For instance, the intelligent water drops (IWDs) metaheuristic has been used to deal with robot path planning problems [124]. However, there are currently only a few examples of such methods being used on ocean unmanned vessels, which can be a future research direction in this field.

Despite the rapid development of ASV path planning, there are still some problems in current studies, which are demonstrated as follows.

1. Defects of Traditional Algorithms

Although the path planning methods for ASVs have experienced significant growth over the last decade, there are still a lot of problems for traditional algorithms that still need to be solved. For instance, the traditional ant colony algorithm has slow convergence speed, the particle swarm algorithm easily falls into local optimality, and the Dijkstra algorithm often generates numerous unnecessary inflection points, which causes a longer distance for the ASV to travel. The use of learning-based algorithms may effectively solve these problems.

2. Problems with Existing Learning-Based Algorithms

Most of the parameters of learning-based algorithms are obtained from experience, so they are affected by subjective factors [71]. Thus, the algorithm's resulting path may need to be revised. To solve this problem, more reasonable and explainable methods of selecting parameters need to be studied so that the obtained learning-based algorithms can perform better.

3. Lack of Multi-ASV Collaborative Path Planning Algorithms

Currently, most path planning algorithms are designed for a single ASV. In intricate missions, deploying multiple ASVs is increasingly preferred over relying on a single one to tackle complex tasks. However, despite their numerous potential applications, coordinating the motion control of ASVs presents significant challenges. These challenges stem from factors such as the multiplicity of ASVs, the complexity of vehicle interactions, the formation of fleets with collision avoidance requirements, and the limited communication bandwidth available in marine environments [125]. The path planning methods of multi-ASV collaboration are still very lacking.

4. Few Studies Based on Dynamic Maps

Most of the current path planning algorithms are obtained using static maps. Research has yet to be carried out on motion variation and uncertainty for dynamic obstacles such as sailing boats, especially in complex waters. Studies based on complex water environments with dynamic maps should be carried out so that moving obstacles and other boats can be considered and the algorithms can be more applicable when used in real marine environments.

5. Lack of Combination with Ship Models

In numerous ASV path planning studies, the ASV is commonly simplified as a point, which could lead to impractical outcomes and potential safety issues when applied in real-world scenarios, particularly in complex marine environments. Consequently, there is a pressing need to integrate the research of ship models and path planning in future studies. By considering the detailed characteristics and dynamics of ASVs within the context of path planning algorithms, researchers can develop more accurate and reliable solutions tailored to the challenges posed by real-world maritime operations. This interdisciplinary approach promises to enhance ASV operations' safety, efficiency, and effectiveness across various applications, from navigation in congested waterways to autonomous search and rescue missions.

With the rapid development of technology, the research of ASVs will continue to advance. Below are several directions for future ASV studies.

1. More Learning-based Approaches

With the rapid development of machine learning, learning-based path planning methods for ASVs emerge as a promising avenue poised to attract increased attention in the near future. Using learning-based approaches, problems in traditional path planning algorithms can be solved, significantly improving the performance of path planning for an ASV.

2. Improved Parameter Selection

To reduce the impact of subjective factors, the methods of selecting parameters in learning-based algorithms should be improved. Some reasonable methods can be used for parameter selection. For example, the grid search method systematically explores a predefined set of parameter values over a grid. It selects the best performance combination based on some evaluation metric. Bayesian optimization uses probabilistic models to predict the performance of different parameter configurations based on past evaluations and guide the search toward promising regions of the parameter space. The obtained learning-based algorithms can perform better using more explainable parameter selection methods.

3. Collaborative Path Planning

In complex missions, there is a growing demand to deploy multiple ASVs instead of relying on a single one to accomplish challenging tasks. Cooperative operations among a fleet of ASVs provide significant advantages, enhancing capability and effectiveness [126]. In the future, it is necessary to investigate collaborative path planning strategies for multiple ASVs operating in the same area to optimize fleet coordination and achieve complex mission objectives.

4. More Research Based on Dynamic Maps

To obtain more practical ASV path planning algorithms, using dynamic maps to simulate real marine environments is significant. In real situations, the changing environment with moving obstacles and other ships may affect the result of path planning algorithms, which are studied based on static maps. Thus, more research based on dynamic maps is a trend for future study.

5. Combination with Ship Models

Integrating path planning research with the study of ship models is crucial for achieving more accurate results. Safety concerns such as collisions can be mitigated by incorporating detailed ship models instead of treating boats as mere points. This approach enables a more comprehensive understanding of vessel dynamics and environmental interactions, thereby enhancing the precision and safety of path planning algorithms in diverse maritime contexts.

6. More Intelligent ASV Path Planning

In the future, creating more intelligent path planning methods for ASV is a promising direction. There are three dimensions to be considered. Firstly, ASV path planning algorithms should be integrated with existing navigation systems and maritime traffic management frameworks to ensure compatibility and interoperability with other vessels and maritime infrastructure. In addition, collision avoidance mechanisms and safety protocols should be enhanced to ensure the safe operation of ASVs, particularly in crowded or dynamic maritime environments. Last but not least, advanced ASV path planning algorithms should be used to enable autonomous decision-making capabilities, allowing

ASVs to navigate and adapt to unforeseen circumstances autonomously without human intervention.

ASVs are anticipated to reshape the landscape of future sea warfare, potentially serving as disruptive assets in the ongoing struggle for maritime supremacy, and play a massive role in the implementation of reconnaissance and surveillance, anti-submarine warfare, anti-mine warfare, logistics support, etc. Also, it will improve operational efficiency and reduce casualties.

From a civilian perspective, ASVs offer a cost-effective solution for maritime patrol and surveillance, providing timely feedback and extended monitoring capabilities even in challenging conditions. They also serve as a rapid, efficient, and economical platform for broad search and rescue operations. Through their mobility, ASVs can provide comprehensive insights into the distribution and spread of pollution in rivers, lakes, and oceans, paving the way for innovative approaches to water environment management [127].

Also, extensive field testing and validation of ASV path planning algorithms in real-world maritime environments are essential to assess the proposed algorithms' performance and reliability.

5. Conclusions

This review begins by outlining the problem definition and assessment criteria for path planning in ASVs. It then categorizes and reviews various path planning algorithms in the literature, specifically focusing on unmanned vessels and sailboats, discussing their strengths and weaknesses alongside evaluation indicators. Additionally, this paper explores how marine environmental conditions impact path planning and the strategies used to address these challenges. Ultimately, this review highlights the obstacles faced in ASV path planning and suggests possible technological solutions and future research directions, serving as a reference for further studies in this area.

Author Contributions: Methodology, Y.W.; investigation, Y.W.; writing—original draft preparation, Y.W.; writing—review and editing, T.W. and S.L.; supervision, T.W. and S.L; funding acquisition, T.W. and S.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the "Pioneer" and "Leading Goose" R&D Program of Zhejiang (2023C03124), and the research project of Robotics Institute, Zhejiang University (109107-Y11801).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: No new data is created.

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

References

1. Liu, C.; Chu, X.; Wu, Q.; Wang, G.-C. Current status and prospects of USV. *Shipbuild. China* **2014**, *55*, 194–205.
2. Bertram, V. *Unmanned Surface Vehicles—A Survey*; Skibsteknisk Selskab: Copenhagen, Denmark, 2008; Volume 1, pp. 1–4.
3. Mou, P. *Research on Trajectory Tracking Control Method of Surface Unmanned Ship*; Harbin Engineering University: Harbin, China, 2014.
4. Jing, W.; Liu, C.; Li, T.; Rahman, A.M.; Xian, L.; Wang, X.; Wang, Y.; Guo, Z.; Brenda, G.; Tendai, K.W. Path planning and navigation of oceanic autonomous sailboats and vessels: A survey. *J. Ocean Univ. China* **2020**, *19*, 609–621. [CrossRef]
5. Wei, F. *A Study on Path Planning and Tracking Control of Unmanned Surface Vehicle*; Harbin Engineering University: Harbin, China, 2016.
6. Liu, W.; Tan, G.; Zou, J.; Dong, C.; Tang, Z. A Study on Path Planning of Unpiloted Vehicles Based on Collision Avoidance Rules. *J. Inf. Control.* **2020**, *50*, 308–320.
7. Cheng, C.; Sha, Q.; He, B.; Li, G. Path planning and obstacle avoidance for AUV: A review. *Ocean Eng.* **2021**, *235*, 109355. [CrossRef]
8. Er, M.J.; Ma, C.; Liu, T.; Gong, H. Intelligent motion control of unmanned surface vehicles: A critical review. *Ocean Eng.* **2023**, *280*, 114562. [CrossRef]
9. Xiang, L.I.; Xiaoming, Y.E.; Quanbin, W.A.; Weiguang, L.I.; Hanlin, G.A. Review on the research of local path planning algorithms for unmanned surface vehicles. *Chin. Ship Res.* **2021**, *16* (Suppl. S1), 1–10.

10. 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]
11. Wu, G.; Xu, T.; Sun, Y.; Zhang, J. Review of multiple unmanned surface vessels collaborative search and hunting based on swarm intelligence. *Int. J. Adv. Robot. Syst.* **2022**, *19*, 17298806221091885. [CrossRef]
12. Sanchez-Ibanez, J.R.; Perez-del-Pulgar, C.J.; García-Cerezo, A. Path planning for autonomous mobile robots: A review. *Sensors* **2021**, *21*, 7898. [CrossRef]
13. Souissi, O.; Benatitallah, R.; Duvivier, D.; Artiba, A.; Belanger, N.; Feyzeau, P. Path planning: A 2013 survey. In Proceedings of the 2013 International Conference on Industrial Engineering and Systems Management (IESM), Agdal, Morocco, 28 October 2013; pp. 1–8.
14. Nash, A.; Koenig, S. Any-angle path planning. *AI Mag.* **2013**, *34*, 85–107. [CrossRef]
15. Algfoor, Z.A.; Sunar, M.S.; Kolivand, H. A comprehensive study on pathfinding techniques for robotics and video games. *Int. J. Comput. Games Technol.* **2015**, *2015*, 7. [CrossRef]
16. Petres, C.; Pailhas, Y.; Petillot, Y.; Lane, D. Underwater path planing using fast marching algorithms. In *Europe Oceans 2005*; IEEE: Piscataway, NJ, USA, 2005; Volume 2, pp. 814–819.
17. Barraquand, J.; Langlois, B.; Latombe, J.C. Numerical potential field techniques for robot path planning. *IEEE Trans. Syst. Man Cybern.* **1992**, *22*, 224–241. [CrossRef]
18. Campbell, S.; Naeem, W.; Irwin, G.W. A review on improving the autonomy of unmanned surface vehicles through intelligent collision avoidance manoeuvres. *Annu. Rev. Control* **2012**, *36*, 267–283. [CrossRef]
19. Kim, H.; Kim, D.; Shin, J.U.; Kim, H.; Myung, H. Angular rate-constrained path planning algorithm for unmanned surface vehicles. *Ocean Eng.* **2014**, *84*, 37–44. [CrossRef]
20. Kim, H.; Kim, D.; Kim, H.; Shin, J.U.; Myung, H. An extended any-angle path planning algorithm for maintaining formation of multi-agent jellyfish elimination robot system. *Int. J. Control Autom. Syst.* **2016**, *14*, 598–607. [CrossRef]
21. Wang, N.; Su, S.F. Finite-time unknown observer-based interactive trajectory tracking control of asymmetric underactuated surface vehicles. *IEEE Trans. Control Syst. Technol.* **2019**, *29*, 794–803. [CrossRef]
22. Cheng, Y.; Zhang, W. Concise deep reinforcement learning obstacle avoidance for underactuated unmanned marine vessels. *Neurocomputing* **2018**, *272*, 63–73. [CrossRef]
23. Lazarowska, A. A new deterministic approach in a decision support system for ship's trajectory planning. *Expert Syst. Appl.* **2017**, *71*, 469–478. [CrossRef]
24. Peralta, F.; Arzamendia, M.; Gregor, D.; Reina, D.G.; Toral, S. A comparison of local path planning techniques of autonomous surface vehicles for monitoring applications: The ypacarai lake case-study. *Sensors* **2020**, *20*, 1488. [CrossRef]
25. Cruz, N.A.; Alves, J.C. Autonomous sailboats: An emerging technology for ocean sampling and surveillance. In *Oceans 2008*; IEEE: Piscataway, NJ, USA, 2008; pp. 1–6.
26. Kaur, P.; Aziz, A.; Jain, D.; Patel, H.; Hirokawa, J.; Townsend, L.; Reimers, C.; Hua, F. Sea situational awareness (seasaw) dataset. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition 2022, New Orleans, LA, USA, 18–24 June 2022; pp. 2579–2587.
27. Koutantou, K.; Brunner, P.; Vazquez-Cuervo, J. Validation of NASA Sea Surface Temperature Satellite Products Using Saildrone Data. *Remote Sens.* **2023**, *15*, 2277. [CrossRef]
28. Friebe, A.; Olsson, M.; Le Gallic, M.; Springett, J.L.; Dahl, K.; Waller, M. A marine research ASV utilizing wind and solar power. In Proceedings of the OCEANS 2017, Aberdeen, UK, 19–22 June 2017; pp. 1–7.
29. Dijkstra, E.W. A note on two problems in connexion with graphs. In *Edsger Wybe Dijkstra: His Life, Work, and Legacy*; ACM Books: New York, NY, USA, 2022; pp. 287–290.
30. Zhu, Z.; Li, L.; Wu, W.; Jiao, Y. Application of improved Dijkstra algorithm in intelligent ship path planning. In Proceedings of the 2021 33rd Chinese Control and Decision Conference (CCDC), Kunming, China, 22–24 May 2021; pp. 4926–4931.
31. Hart, P.E.; Nilsson, N.J.; Raphael, B. A formal basis for the heuristic determination of minimum cost paths. *IEEE Trans. Syst. Sci. Cybern.* **1968**, *4*, 100–107. [CrossRef]
32. Singh, Y.; Sharma, S.; Sutton, R.; Hatton, D.; Khan, A. A constrained A* approach towards optimal path planning for an unmanned surface vehicle in a maritime environment containing dynamic obstacles and ocean currents. *Ocean Eng.* **2018**, *169*, 187–201. [CrossRef]
33. Yu, J.; Deng, W.; Zhao, Z.; Wang, X.; Xu, J.; Wang, L.; Sun, Q.; Shen, Z. A hybrid path planning method for an unmanned cruise ship in water quality sampling. *IEEE Access* **2019**, *7*, 87127–87140. [CrossRef]
34. Chen, L.; Negenborn, R.R.; Lodewijks, G. Path planning for autonomous inland vessels using A* BG. In Proceedings of the Computational Logistics: 7th International Conference, ICCL 2016, Proceedings 7, Lisbon, Portugal, 7–9 September 2016; Springer: Berlin/Heidelberg, Germany; pp. 65–79.
35. Gu, S.; Zhou, C.; Wen, Y.; Zhong, X.; Zhu, M.; Xiao, C.; Du, Z. A motion planning method for unmanned surface vehicle in restricted waters. *Proc. Inst. Mech. Eng. Part M J. Eng. Marit. Environ.* **2020**, *234*, 332–345. [CrossRef]
36. Xu, S.; Li, W.; Zhang, Z.; Wu, J.; Long, C. Global Path Planning for Unmanned Surface Vessels based on Risk-Penalty-related A* Algorithm. In Proceedings of the 2022 41st Chinese Control Conference (CCC), Hefei, China, 25–27 July 2022; pp. 2422–2427.
37. Wengang, L.I.; Liujiang, W.A.; Dexiang, F.A.; Yuwei, L.I.; Jun, H. Path planning algorithm combining A* with DWA. *Syst. Eng. Electron.* **2021**, *43*, 3694.

38. Tao, L. Research on path planning of unmanned ship based on improved A* algorithm. *Ship Sci. Technol.* **2022**, *44*, 134–137.
39. Dorigo, M. Optimization, Learning and Natural Algorithms. Ph.D. Thesis, Politecnico di Milano, Milano, Italy, 1992.
40. Jialin, L.I.; Jianqiang, Z.H. Global path planning of unmanned boat based on improved ant colony algorithm. In Proceedings of the 2021 4th International Conference on Electron Device and Mechanical Engineering (ICEDME), Guangzhou, China, 19–21 March 2021; pp. 176–179.
41. Liu, X.; Li, Y.; Zhang, J.; Zheng, J.; Yang, C. Self-adaptive dynamic obstacle avoidance and path planning for USV under complex maritime environment. *IEEE Access* **2019**, *7*, 114945–114954. [CrossRef]
42. Xinchi, T.; Huajun, Z.; Wenwen, C.; Peimin, Z.; Zhiwen, L.; Kai, C. A research on intelligent obstacle avoidance for unmanned surface vehicles. In Proceedings of the 2018 Chinese Automation Congress (CAC), Xi'an, China, 30 November–2 December 2018; pp. 1431–1435.
43. Yuwen C, Zhao X, Autonomous navigation path planning of unmanned ship based on hybrid ant colony algorithm. *Ship Sci. Technol.* **2023**, *45*, 93–96.
44. Tong, W.; Kai, M. Path planning of unmanned ships in restricted waters based on improved ant colony algorithm. *Ship Electron. Eng.* **2023**, *43*, 120–124+128.
45. Shouqi, M.; Ping, Y.; Diju, G. Path planning of unmanned ships based on bacteria forage-ant colony algorithm. *Control Eng.* **2024**, *31*, 1–9. [CrossRef]
46. Kennedy, J.; Eberhart, R. Particle swarm optimization. In Proceedings of the ICNN'95—International conference on neural networks, Perth, Australia, 27 November–1 December 1995; Volume 4, pp. 1942–1948.
47. Verma, A.; Dhanda, N.; Yadav, V. Binary particle swarm optimization based edge detection under weighted image sharpening filter. *Int. J. Inf. Technol.* **2023**, *15*, 289–299. [CrossRef]
48. Liu, C.; Liu, K. Global Path Planning for Unmanned Ships Based on Improved Particle Swarm Algorithm. In Proceedings of the International Conference on Bio-Inspired Computing: Theories and Applications, Wuhan, China, 16–18 December 2022; Springer Nature: Singapore, 2023; pp. 106–116.
49. Zhong, J.; Li, B.; Li, S.; Yang, F.; Li, P.; Cui, Y. Particle swarm optimization with orientation angle-based grouping for practical unmanned surface vehicle path planning. *Appl. Ocean Res.* **2021**, *111*, 102658. [CrossRef]
50. Gao, B.; Wang, L.; Zhang, S.; Chen, J. On improvement of path planning of unmanned ship using particle swarm optimization algorithm. *Ship Boat* **2022**, *33*, 47.
51. Gong, Y.; Luo, M.; Wang, C.; Han, T. A path planning method based on improved particle swarm optimization algorithm. In Proceedings of the 2020 7th International Conference on Information Science and Control Engineering (ICISCE), Changsha, China, 18–20 December 2020; pp. 106–110.
52. Shin, J.; Kwak, D.J.; Lee, Y.I. Adaptive path-following control for an unmanned surface vessel using an identified dynamic model. *IEEE/ASME Trans. Mechatron.* **2017**, *22*, 1143–1153. [CrossRef]
53. Song, L.; Chen, H.; Xiong, W.; Dong, Z.; Mao, P.; Xiang, Z.; Hu, K. Method of emergency collision avoidance for unmanned surface vehicle (USV) based on motion ability database. *Pol. Marit. Res.* **2019**, *2*, 55–67. [CrossRef]
54. Xin, J.; Zhong, J.; Li, S.; Sheng, J.; Cui, Y. Greedy mechanism based particle swarm optimization for path planning problem of an unmanned surface vehicle. *Sensors* **2019**, *19*, 4620. [CrossRef] [PubMed]
55. Wang, C.; Mao, Y.S.; Du, K.J.; Hu, B.Q.; Song, L.F. Simulation on local obstacle avoidance algorithm for unmanned surface vehicle. *Int. J. Simul. Model.* **2016**, *15*, 460–472. [CrossRef] [PubMed]
56. Shuping, G. *Unmanned Surface Vehicle Path Planning Based on Improved Particle-Artificial Bee Colony Hybrid Algorithm*; Dalian Maritime University: Dalian, China, 2022. [CrossRef]
57. Sethian, J.A. Fast marching methods. *SIAM Rev.* **1999**, *41*, 199–235. [CrossRef]
58. Liu, Y.; Bucknall, R. Path planning algorithm for unmanned surface vehicle formations in a practical maritime environment. *Ocean Eng.* **2015**, *97*, 126–144. [CrossRef]
59. 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]
60. Naitian, Z.; Shicai, C.; Zixin, M. Global path planning of surface unmanned ship based on improved fast travel method. *J. Shanghai Marit. Univ.* **2023**, *44*, 5–11. [CrossRef]
61. LaValle, S. Rapidly-Exploring Random Trees: A New Tool for Path Planning. The Annual Research Report 9811. **1998**. Available online: https://api.semanticscholar.org/CorpusID:14744621 (accessed on 20 March 2024).
62. Yu, J.; Chen, Z.; Zhao, Z.; Yao, P.; Xu, J. A traversal multi-target path planning method for multi-unmanned surface vessels in space-varying ocean current. *Ocean Eng.* **2023**, *278*, 114423. [CrossRef]
63. Xie, X.; Wang, Y.; Wu, Y.; You, M.; Zhang, S. Random Patrol Path Planning for Unmanned Surface Vehicles in Shallow Waters. In Proceedings of the 2022 IEEE International Conference on Mechatronics and Automation (ICMA), Guilin, China, 7–10 August 2022; pp. 1860–1865.
64. Ouyang, Z.; Wang, H.; Huang, Y.; Yang, K.; Yi, H. Path planning technologies for USV formation based on improved RRT. *Chin. J. Ship Res.* **2020**, *15*, 18–24.
65. Cao, S.; Fan, P.; Yan, T.; Xie, C.; Deng, J.; Xu, F.; Shu, Y. Inland waterway ship path planning based on improved RRT algorithm. *J. Mar. Sci. Eng.* **2022**, *10*, 1460. [CrossRef]

66. Lin, Y.; Zhang, W.; Mu, C.; Wang, J. Application of improved RRT algorithm in unmanned surface vehicle path planning. In Proceedings of the 2022 34th Chinese Control and Decision Conference (CCDC), Hefei, China, 15–17 August 2022; pp. 4861–4865.
67. Sheng, Q. *Research on Unmanned Ship Path Planning Based on Improved Fast Extended Random Tree Method*; Dalian Ocean University: Dalian, China, 2024.
68. Da, Z. *Research on Surface Target Recognition and Path Planning of Unmanned Ship*; Jilin University: Jilin, China, 2024.
69. Xichao, Z.; Yong, Y. Unmanned ship path planning based on improved RRT* algorithm. *Chin. Navig.* **2023**, *46*, 143–147+154.
70. Koenig, S.; Likhachev, M. D* lite. In Proceedings of the Eighteenth National Conference on Artificial intelligence, Edmonton, AB, Canada, 28 July–1 August 2002; pp. 476–483.
71. 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]
72. Chen, H.; Wu, Z.; Zheng, R.; Zhang, S. Design of Autonomous Obstacle Avoidance Unmanned Boat System for Wetland Monitoring. *J. Phys. Conf. Ser.* **2020**, *1486*, 072033. [CrossRef]
73. Guo, S.; Zhang, X.; Zheng, Y.; Du, Y. An autonomous path planning model for unmanned ships based on deep reinforcement learning. *Sensors* **2020**, *20*, 426. [CrossRef]
74. Wang, C.; Zhang, X.; Li, R.; Dong, P. Path planning of maritime autonomous surface ships in unknown environment with reinforcement learning. In Proceedings of the Cognitive Systems and Signal Processing: 4th International Conference, ICCSIP 2018, Beijing, China, 29 November–1 December 2018; Revised Selected Papers, Part II 4 2019. Springer: Singapore, 2019; pp. 127–137.
75. Chen, C.; Chen, X.Q.; Ma, F.; Zeng, X.J.; Wang, J. A knowledge-free path planning approach for smart ships based on reinforcement learning. *Ocean Eng.* **2019**, *189*, 106299. [CrossRef]
76. Du, Y.; Zhang, X.; Cao, Z.; Wang, S.; Liang, J.; Zhang, F.; Tang, J. An optimized path planning method for coastal ships based on improved DDPG and DP. *J. Adv. Transp.* **2021**, *2021*, 1–23. [CrossRef]
77. Li, L.; Wu, D.; Huang, Y.; Yuan, Z.M. A path planning strategy unified with a COLREGS collision avoidance function based on deep reinforcement learning and artificial potential field. *Appl. Ocean Res.* **2021**, *113*, 102759. [CrossRef]
78. Zhao, J.; Wang, P.; Li, B.; Bai, C. A DDPG-Based USV Path-Planning Algorithm. *Appl. Sci.* **2023**, *13*, 10567. [CrossRef]
79. 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]
80. Wu, J.; Cheng, L.; Chu, S.; Song, Y. An autonomous coverage path planning algorithm for maritime search and rescue of persons-in-water based on deep reinforcement learning. *Ocean Eng.* **2024**, *291*, 116403. [CrossRef]
81. McCulloch, W.S.; Pitts, W. A logical calculus of the ideas immanent in nervous activity. *Bull. Math. Biophys.* **1943**, *5*, 115–133. [CrossRef]
82. Wang, R.; Miao, K.; Li, Q.; Sun, J.; Deng, H. The path planning of collision avoidance for an unmanned ship navigating in waterways based on an artificial neural network. *Nonlinear Eng.* **2022**, *11*, 680–692. [CrossRef]
83. Plumet, F.; Petres, C.; Romero-Ramirez, M.A.; Gas, B.; Ieng, S.H. Toward an autonomous sailing boat. *IEEE J. Ocean. Eng.* **2014**, *40*, 397–407. [CrossRef]
84. Shen, Z.; Ding, W.; Liu, Y.; Yu, H. Path planning optimization for unmanned sailboat in complex marine environment. *Ocean Eng.* **2023**, *269*, 113475. [CrossRef]
85. Jianyun, X.; Jingsong, X.; Chunxiao, H. Autonomous Obstacle Avoidance Algorithm for Unmanned Sailing Ships. *Ship Eng.* **2018**, *40*, 61–65. [CrossRef]
86. Jianyun, X. *Research on Obstacle Avoidance System of Unmanned Sailboat*; Shanghai Jiao Tong University: Shanghai, China, 2020. [CrossRef]
87. Shaolong, Y. *Research on Unmanned Sailing Boat Design and Motion Control*; Dalian Maritime University: Dalian, China, 2013.
88. Langbein, J.; Stelzer, R.; Frühwirth, T. A rule-based approach to long-term routing for autonomous sailboats. In *Robotic Sailing, Proceedings of the 4th International Robotic Sailing Conference, Lübeck, Germany, 15 July 2011*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 195–204.
89. Mannarini, G.; Lecci, R.; Coppini, G. Introducing sailboats into ship routing system VISIR. In Proceedings of the 2015 6th International Conference on Information, Intelligence, Systems and Applications (IISA), Corfu, Greece, 6–8 July 2015; pp. 1–6.
90. Saoud, H.; Hua, M.D.; Plumet, F.; Amar, F.B. Routing and course control of an autonomous sailboat. In Proceedings of the 2015 European Conference on Mobile Robots (ECMR), Lincoln, UK, 2–4 September 2015; pp. 1–6.
91. Nian, R.; Shen, Z.; Ding, W. Research on global path planning of unmanned sailboat based on improved ant colony optimization. In Proceedings of the 2021 6th International Conference on Automation, Control and Robotics Engineering (CACRE) 2021, Dalian, China, 15–17 July 2021; pp. 428–432.
92. Zhipeng, S.; Wenna, D.; Yuchen, L. Unmanned sailboat path planning based on improved adaptive ant colony algorithm. *J. Harbin Eng. Univ.* **2023**, *44*, 969–976.
93. Runda, N. *Research on Path Planning of Unmanned Sailboat with Wind Field Constraints*; Dalian Maritime University: Dalian, China, 2021. [CrossRef]
94. Li, P.; Wang, H.; Li, X. Improved ant colony algorithm for global path planning. *AIP Conf. Proc.* **2017**, *1820*, 080013.
95. Khatib, O. The potential field approach and operational space formulation in robot control. In *Adaptive and Learning Systems: Theory and Applications*; Springer: Boston, MA, USA, 1986; pp. 367–377.

96. Pêtrès, C.; Romero-Ramirez, M.A.; Plumet, F. Reactive path planning for autonomous sailboat. In Proceedings of the 2011 15th International Conference on Advanced Robotics (ICAR), Tallinn, Estonia, 20–23 June 2011; pp. 112–117.
97. Plumet, F.; Saoud, H.; Hua, M.D. Line following for an autonomous sailboat using potential fields method. In Proceedings of the 2013 MTS/IEEE OCEANS, Bergen, Germany, 10 June 2013; pp. 1–6.
98. Romero-Ramirez, M.A.; Petres, C.; Plumet, F. Navigation with obstacle avoidance of an autonomous sailboat. In Proceedings of the 14th International Conference on Climbing and Walking Robots (Clawar2011), Miguel, Portugal, 12–14 September 2011; pp. 86–93.
99. Wentao, Z.; Longyue, J.; Chaoyang, S. Dynamic Collision Avoidance Method of Unmanned Sailboat Based on Improved Fast Leveling Method. *Chin. Ship.* **2023**, *19*, 1–14.
100. Tianyuan, X.; Yanwei, Z.; Hua, T. Research on Real-time Path Planning of Unmanned Sailing Ships Based on Probability Potential Field. *Foreign Electron. Meas. Technol.* **2020**, *39*, 17–22. [CrossRef]
101. Yuqin, D. *Unmanned Sailing Boat Behavior Learning and Trajectory Tracking Control and Planning*; Donghua University: Donghua, China, 2024.
102. Silva Junior, A.G.; Santos, D.H.; Negreiros, A.P.; Silva, J.M.; Gonçalves, L.M. High-level path planning for an autonomous sailboat robot using Q-Learning. *Sensors* **2020**, *20*, 1550. [CrossRef] [PubMed]
103. Barrera, C.; Maarouf, M.; Campuzano, F.; Llinas, O.; Marichal, G.N. A comparison of intelligent models for collision avoidance path planning on environmentally propelled unmanned surface vehicles. *J. Mar. Sci. Eng.* **2023**, *11*, 692. [CrossRef]
104. Bitar, G.; Martinsen, A.B.; Lekkas, A.M.; Breivik, M. Two-stage optimized trajectory planning for ASVs under polygonal obstacle constraints: Theory and experiments. *IEEE Access* **2020**, *8*, 199953–199969. [CrossRef]
105. Zhao, L.; Wang, F.; Bai, Y. Route planning for autonomous vessels based on improved artificial fish swarm algorithm. *Ships Offshore Struct.* **2023**, *18*, 897–906. [CrossRef]
106. Niu, H.; Lu, Y.; Savvaris, A.; Tsourdos, A. Efficient path planning algorithms for unmanned surface vehicle. *IFAC-PapersOnLine* **2016**, *49*, 121–126. [CrossRef]
107. Liu, C.; Mao, Q.; Chu, X.; Xie, S. An improved A-star algorithm considering water current, traffic separation and berthing for vessel path planning. *Appl. Sci.* **2019**, *9*, 1057. [CrossRef]
108. Liu, L.; Wang, X.; Yang, X.; Liu, H.; Li, J.; Wang, P. Path planning techniques for mobile robots: Review and prospect. *Expert Syst. Appl.* **2023**, *227*, 120254. [CrossRef]
109. Nian, J. *Indoor Map Construction and Navigation of Mobile Robot Based on Multisensor Fusion*; Beijing University of Posts and Telecommunications: Beijing, China, 2021. [CrossRef]
110. Tian, S.; Li, Y.; Kang, Y.; Xia, J. Multi-robot path planning in wireless sensor networks based on jump mechanism PSO and safety gap obstacle avoidance. *Future Gener. Comput. Syst.* **2021**, *118*, 37–47. [CrossRef]
111. Raaj, Y.; John, A.; Jin, T. 3D Object Localization using Forward Looking Sonar (FLS) and Optical Camera via particle filter based calibration and fusion. In Proceedings of the OCEANS 2016 MTS/IEEE Conference, Monterey, CA, USA, 9–23 September 2016; pp. 1–10.
112. Ji, T.; Sivakumar, A.N.; Chowdhary, G.; Driggs-Campbell, K. Proactive anomaly detection for robot navigation with multi-sensor fusion. *IEEE Robot. Autom. Lett.* **2022**, *7*, 4975–4982. [CrossRef]
113. Han, Y.; Zhan, I.H.; Zhao, W.; Pan, J.; Zhang, Z.; Wang, Y.; Liu, Y.J. Deep reinforcement learning for robot collision avoidance with self-state-attention and sensor fusion. *IEEE Robot. Autom. Lett.* **2022**, *7*, 6886–6893. [CrossRef]
114. Henkel, C.; Bubeck, A.; Xu, W. Energy efficient dynamic window approach for local path planning in mobile service robotics. *IFAC-PapersOnLine* **2016**, *49*, 32–37. [CrossRef]
115. 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]
116. Jin, Q.; Tang, C.; Cai, W. Research on dynamic path planning based on the fusion algorithm of improved ant colony optimization and rolling window method. *IEEE Access* **2021**, *10*, 28322–28332. [CrossRef]
117. Chang, L.; Shan, L.; Jiang, C.; Dai, Y. Reinforcement based mobile robot path planning with improved dynamic window approach in unknown environment. *Auton. Robots* **2021**, *45*, 51–76. [CrossRef]
118. Liu, C.; Lee, S.; Varnhagen, S.; Tseng, H.E. Path planning for autonomous vehicles using model predictive control. In Proceedings of the 2017 IEEE Intelligent Vehicles Symposium (IV), Los Angeles, CA, USA, 11–14 June 2017; pp. 174–179.
119. Li, S.; Li, Z.; Yu, Z.; Zhang, B.; Zhang, N. Dynamic trajectory planning and tracking for autonomous vehicle with obstacle avoidance based on model predictive control. *IEEE Access* **2019**, *7*, 132074–132086. [CrossRef]
120. Hang, P.; Huang, S.; Chen, X.; Tan, K.K. Path planning of collision avoidance for unmanned ground vehicles: A nonlinear model predictive control approach. *Proc. Inst. Mech. Eng. Part I J. Syst. Control Eng.* **2021**, *235*, 222–236. [CrossRef]
121. Chavez-Galaviz, J.; Li, J.; Chaudhary, A.; Mahmoudian, N. ASV station keeping under wind disturbances using neural network simulation error minimization model predictive control. *J. Field Robot.* **2023**. [CrossRef]
122. Cailian, L.A.; Peng, L.I.; Yu, F.E. Path planning of greenhouse robot based on fusion of improved A* algorithm and dynamic window approach. *Nongye Jixie Xuebao/Trans. Chin. Soc. Agric. Mach.* **2021**, *52*.
123. Zhao, L.; Bai, Y. Data harvesting in uncharted waters: Interactive learning empowered path planning for USV-assisted maritime data collection under fully unknown environments. *Ocean Eng.* **2023**, *287*, 115781. [CrossRef]

124. Darvishpoor, S.; Darvishpour, A.; Escarcega, M.; Hassanalian, M. Nature-inspired algorithms from oceans to space: A comprehensive review of heuristic and meta-heuristic optimization algorithms and their potential applications in drones. *Drones* **2023**, *7*, 427. [CrossRef]

125. Salmanpour, S.; Motameni, H. Optimal path planning for mobile robot using Intelligent Water Drops algorithm. *J. Intell. Fuzzy Syst.* **2014**, *27*, 1519–1531. [CrossRef]

126. Peng, Z.; Wang, J.; Wang, D.; Han, Q.L. An overview of recent advances in coordinated control of multiple autonomous surface vehicles. *IEEE Trans. Ind. Inform.* **2020**, *17*, 732–745. [CrossRef]

127. Xue, S.; Wang, Q.; Xiao, J.; Liu, J.; Xiong, J. Overview of unmanned surface vessel coverage path planning techniques. In Proceedings of the AIIPCC 2021: Second International Conference on Artificial Intelligence, Information Processing and Cloud Computing, Hangzhou, China, 26–28 June 2021; pp. 1–7.

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.