Journal of *Marine Science and Engineering*

MDPI

*Article*

# Hybrid Probabilistic Road Map Path Planning for Maritime Autonomous Surface Ships Based on Historical AIS Information and Improved DP Compression

**Gongxing Wu [1,*], Liepan Guo [2], Danda Shi [1], Bing Han [3] and Fan Yang [1]**

1 College of Ocean Science and Engineering, Shanghai Maritime University, Shanghai 201306, China; ddshi@shmtu.edu.cn (D.S.); 202430410057@stu.shmtu.edu.cn (F.Y.)
2 Merchant Marine College, Shanghai Maritime University, Shanghai 201306, China; liepanguo@gmail.com
3 Shanghai Ship and Shipping Research Institute Co., Ltd., Shanghai 200135, China; han.bing@coscoshipping.com
* Correspondence: wugx@shmtu.edu.cn

**Abstract:** A hybrid probabilistic road map (PRM) path planning algorithm based on historical automatic identification system (AIS) information and Douglas–Peucker (DP) compression is proposed to address the issues of low path quality and the need for extensive sampling in the traditional PRM algorithm. This innovative approach significantly reduces the number of required samples and decreases path planning time. The process begins with the collection of historical AIS data from the autonomous vessel's navigation area, followed by comprehensive data-cleaning procedures to eliminate invalid and incomplete records. Subsequently, an enhanced DP compression algorithm is employed to streamline the cleaned AIS data, minimizing waypoint data while retaining essential trajectory characteristics. Intersection points among various vessel trajectories are then calculated, and these points, along with the compressed AIS data, form the foundational dataset for path searching. Building upon the traditional PRM framework, the proposed hybrid PRM algorithm integrates the B-spline algorithm to smooth and optimize the generated paths. Comparative experiments conducted against the standard PRM algorithm, A*, and Dijkstra algorithms demonstrate that the hybrid PRM approach not only reduces planning time but also achieves superior path smoothness. These improvements enhance both the efficiency and accuracy of path planning for maritime autonomous surface ships (MASS), marking a significant advancement in autonomous maritime navigation.

**Keywords:** AIS historical data; ship path planning; improved DP compression algorithm; hybrid probabilistic route map (HPRM) algorithm; maritime autonomous surface ships

check for updates

Academic Editors: Dongdong Mu and Yunsheng Fan

Received: 1 December 2024
Revised: 1 January 2025
Accepted: 15 January 2025
Published: 20 January 2025

**Citation:** Wu, G.; Guo, L.; Shi, D.; Han, B.; Yang, F. Hybrid Probabilistic Road Map Path Planning for Maritime Autonomous Surface Ships Based on Historical AIS Information and Improved DP Compression. *J. Mar. Sci. Eng.* **2025**, *13*, 184. https://doi.org/10.3390/jmse13010184



## 1. Introduction

### *1.1. Background*

With the development of artificial intelligence technology, research on maritime autonomous surface ships (MASS) has become a focal point, with path planning algorithms being one of the key technologies for enabling autonomous navigation [1,2]. As global trade continues to expand, the shipping industry increasingly demands efficient and safe path planning. Traditional path planning methods mainly rely on static nautical charts and heuristic rules, which are insufficient for addressing the dynamic factors and complex environments encountered in real-world navigation. In recent years, the automatic identification system (AIS) has become an essential tool for real-time monitoring of vessel movements (as shown in Figure 1) and has been widely adopted [3–5]. AIS data provide not only rich historical navigation data but also valuable resources for researchers to improve and optimize path planning algorithms.

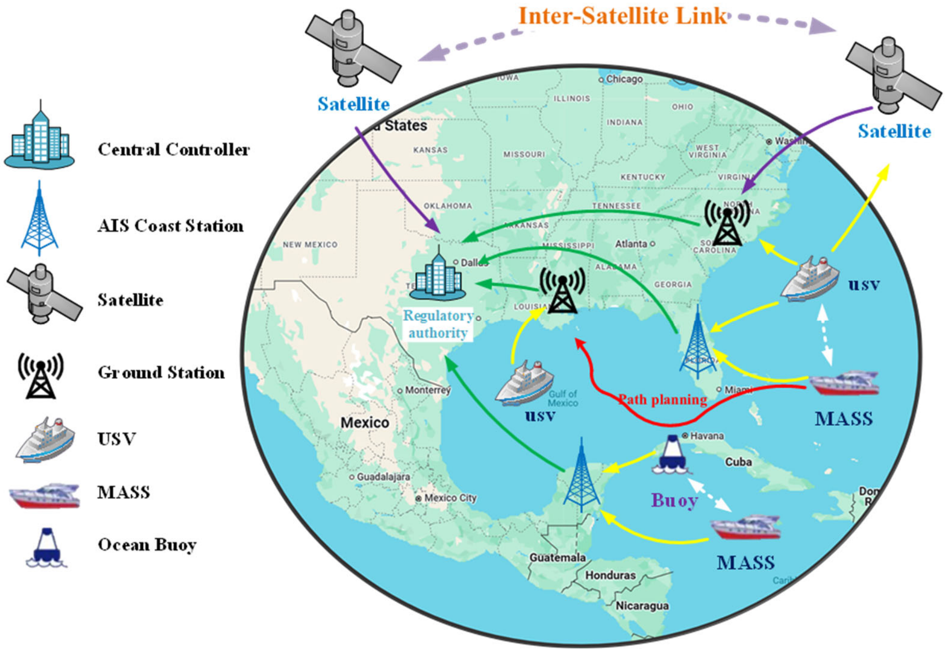



**Figure 1.** Modern maritime communication and navigation system: integrated schematic diagram of the AIS system.

*1.2. Literature Review*

Global maritime path planning algorithms mainly fall into three categories: graph search-based algorithms, sampling-based path planning algorithms, and intelligent optimization algorithms [6–8]. However, in recent years, with the rapid development of computing power, the application of deep reinforcement learning in maritime path planning has become increasingly prevalent [8,9].

Among graph search-based algorithms, they are suitable for static and well-structured environments. The classic shortest path algorithm proposed by the author of the article [10] is widely used, and its variant, the A* algorithm [11–13], is equally popular. Introduced by the author of the article [14], the A* algorithm is an improved heuristic search method that combines the global search capability of Dijkstra's algorithm with heuristic local guidance, thereby enhancing search efficiency.

Intelligent optimization algorithms are methods that mimic natural biological behaviors or social phenomena to solve optimization problems [15,16], such as genetic algorithms [17], particle swarm optimization [18], and ant colony optimization [19]. These algorithms search for optimal solutions in complex search spaces by simulating processes like evolution, group collaboration, or foraging behavior, exhibiting strong global search capabilities and adaptability. However, intelligent optimization algorithms may require substantial computational resources and are sensitive to parameter settings. For example, the authors of the article [20] proposed the APF-ACO algorithm for dynamic and static maritime path planning by combining the artificial potential field method with ant colony optimization. This approach effectively overcomes the shortcomings of traditional path planning methods, though it increases computational complexity. Additionally, while the use of ant colony optimization improves path quality, the smoothness of the paths may still be inferior to some interpolation or optimization-based smoothing methods.

Sampling-based path planning algorithms generate feasible paths by randomly or systematically sampling spatial nodes [21,22], such as rapidly-exploring random trees (RRT) and probabilistic road maps [23,24]. These algorithms are well-suited for complex marine environments, constructing navigable paths by incrementally expanding or connecting sampled points. Compared to traditional global search methods, sampling-based algorithms can more rapidly adapt to dynamic environments and irregular obstacle distributions. However, the generated paths may lack smoothness or optimality, necessitating subsequent optimization to enhance path quality. The authors of the article [25] utilized an improved RRT algorithm combined with AIS data for path planning, followed by a DP algorithm to compress and streamline the path. Although integrating AIS data and DP compression effectively reduces computational load and improves path feasibility, the approach relies heavily on multiple parameter settings, such as the expansion step size of the RRT algorithm and the threshold design and compression tolerance of the DP algorithm. Improper parameter settings may result in unstable paths or reduced computational efficiency.

With the development of data science, data-driven methods have gradually become a research hotspot. The introduction of AIS data provides a substantial amount of actual navigation data, making path planning based on historical data feasible. In recent years, many studies have attempted to utilize machine learning and deep learning techniques to analyze and predict AIS data, thereby enhancing the accuracy of path planning. For instance, the authors of the article [26] proposed a maritime anomaly detection method based on vessel trajectory clustering and classification. This method establishes normal trajectory models using vessel trajectory clustering and then employs a naive Bayes classifier to detect anomalous trajectories. The authors of the article [27] introduced a vessel trajectory prediction method that combines graph attention networks (GAT) with long short-term memory networks (LSTM) to improve the accuracy and robustness of trajectory predictions in the marine engineering field. This approach leverages GAT to capture spatial dependencies in vessel trajectories and integrates LSTM to capture temporal sequence features, thereby achieving efficient and accurate trajectory predictions. Table 1 clearly illustrates the comparison of the advantages and disadvantages of different algorithms.

**Table 1.** Comparison of different path planning algorithms.

| Algorithm | Advantage | Disadvantages |
|---|---|---|
| A* Algorithm | Provides optimal paths with high efficiency | Insufficient consideration for dynamic environments and high computational load |
| Genetic Algorithm | Strong adaptability | Slow convergence speed, no consideration of dynamic environment |
| RRT Algorithm | Suitable for handling dynamic obstacles and real-time path planning | Generates unsmooth paths, and the obtained paths are usually not globally optimal solutions |
| Ship Trajectory Clustering and Classification-Based Method | Utilizes historical data and is capable of detecting anomalous trajectories | Can only detect anomalies and cannot plan paths |
| Method Combining Graph Attention Networks (GAT) and Long Short-Term Memory Networks (LSTM) | Captures spatial and temporal features, improving prediction accuracy and robustness | Computationally complex and requires large amounts of data for training |

However, when handling massive AIS data, how to effectively compress and clean the data remains an urgent problem to be solved by these methods. Moreover, the use of deep learning imposes high demands on computers, requiring extensive computations, and runtime is also an issue.

The DP algorithm, as a classic trajectory compression technique, has been widely applied in AIS data processing in recent years [28–30]. The authors of the article [31] utilized the DP algorithm to compress AIS data, significantly reducing the data volume while preserving key features of the trajectories. However, when processing large-scale data, there is still a trade-off between computational efficiency and compression accuracy in the DP algorithm. Additionally, the path planning process must comprehensively consider the particularities of maritime navigation, such as avoiding land and islands—non-navigable areas—to ensure navigation safety [25,32].

### 1.3. Motivation

At present, although many algorithms are used for ship path planning, their navigation routes are usually based on existing routes rather than new paths generated by these algorithms. Therefore, we plan navigation paths based on historical AIS data to improve the safety of the routes and their adaptability to actual navigation needs. However, AIS data are massive and contain redundant information, making direct use of these data for path planning challenging. Effectively processing and utilizing AIS data has become the key to improving the accuracy and efficiency of path planning. The Douglas–Peucker (DP) algorithm, as a classic trajectory compression technique, can significantly reduce data volume while retaining key features of the trajectories, but it still has certain limitations when processing massive AIS data. Moreover, the path planning process needs to consider the particularities of maritime navigation, such as avoiding land and islands, to ensure navigation safety.

The PRM algorithm, as an effective path planning tool, has important applications in obstacle avoidance and path optimization. Although the traditional PRM algorithm is widely used in path planning, it also has some obvious drawbacks. First, the PRM algorithm relies on randomly sampled nodes; this randomness may make it difficult to cover narrow passages or key areas in complex environments, leading to insufficient connectivity of the generated roadmap or too many path turns. Second, as the number of nodes increases, the computational cost of collision detection rises significantly, especially in high-dimensional spaces, resulting in low algorithm efficiency. Additionally, the paths generated by the traditional PRM algorithm are often not optimal; they may be quite tortuous and require additional smoothing to optimize route quality.

To address this, this study proposes a ship path planning algorithm based on historical AIS information. The algorithm combines an improved DP compression technique and a hybrid PRM algorithm, aiming to efficiently process and utilize AIS data to generate safe and reliable navigation paths. First, AIS data are extracted from the database, and invalid and incomplete records are eliminated through a series of data cleaning processes. Then, the cleaned data are processed using an improved DP compression algorithm, reducing data volume while retaining key features of the trajectories. Subsequently, intersection points among the ship AIS data are calculated, and a hybrid PRM algorithm is developed, which precisely considers the latitude and longitude data of land and islands to ensure that the generated path does not cross or contact these non-navigable areas.

By integrating the improved DP compression technique and the hybrid PRM algorithm, this study not only enhances the efficiency and accuracy of path planning but also fully utilizes historical navigation data, providing new ideas and methods for ship path planning. The proposed method offers strong support for achieving efficient and safe path planning in complex environments within the shipping industry and provides valuable references for related research fields.

### 1.4. Method Overview

In this study, AIS data are utilized to provide initial information for path planning. An improved Douglas–Peucker (DP) trajectory compression algorithm is employed to compress the initial AIS data, and intersection points between different voyages are calculated. Finally, the intersection points and the compressed AIS data are used as navigable points in the ship's path, and an improved PRM algorithm is applied to determine the required path. The proposed method follows the process outlined below:

(1) Extract initial AIS information and perform data cleaning on the initial AIS data to obtain data suitable for compression.
(2) Apply the improved DP algorithm to compress the AIS data, resulting in compressed ship routes.
(3) Compute the latitude and longitude of the intersection points by performing pairwise intersections of the compressed ship routes.
(4) Use the intersection points obtained in the third step along with the compressed AIS latitude and longitude data as navigable points, import environmental information from electronic nautical charts, and finally apply the hybrid PRM algorithm to determine the desired path.

The framework of this process is illustrated in Figure 2:

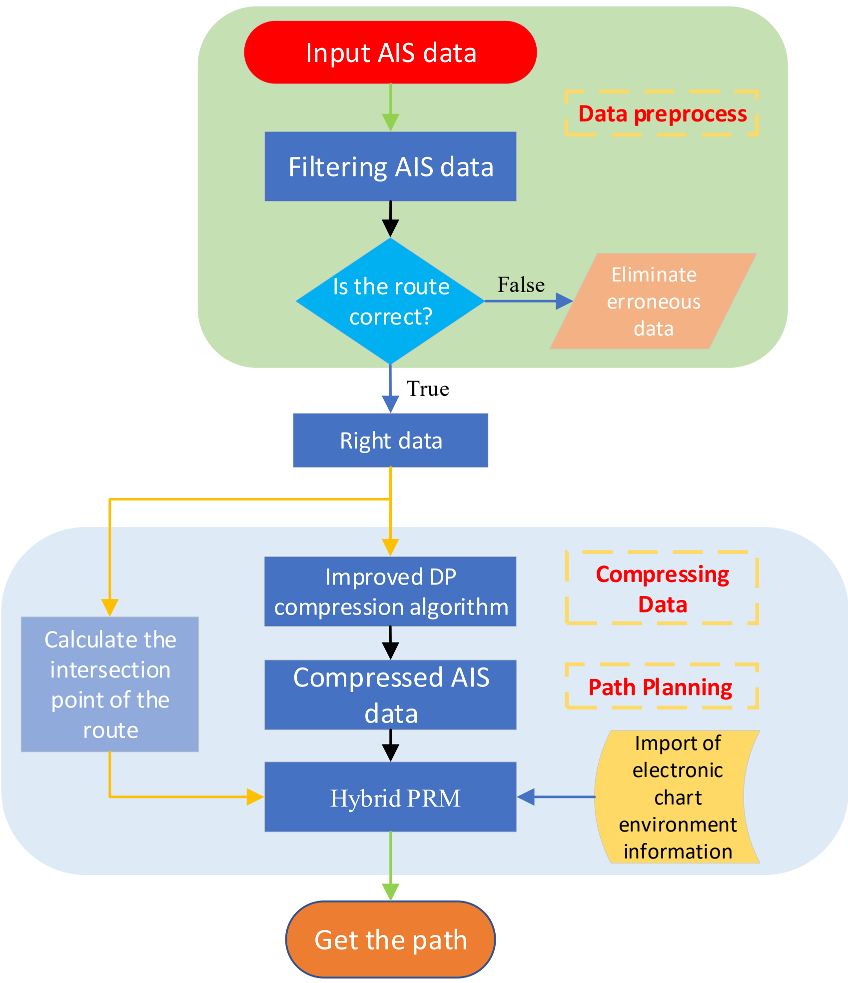



**Figure 2.** Main idea flow chart.

### 1.5. Contributions

Based on the above research background, this study aims to propose an efficient, reliable, and fast path planning method by combining an improved DP compression technique and a HPRM algorithm to cope with the complex environments and dynamic changes in actual navigation.

The main contributions of this paper are as follows:

(1) Proposed a HPRM path planning method that considers historical AIS data and environmental data to plan a new route. In view of the random sampling method of the traditional PRM algorithm, the compressed AIS data and the intersection points between routes are used to replace the random sampling points. At the same time, the electronic nautical chart is analyzed to obtain the longitude and latitude data points of land, coastline, and islands, which will serve as the basis for the environmental map modeling of the HPRM algorithm proposed in this paper;

(2) Utilized an improved DP algorithm to extract concise and consistent features from the original AIS data, thereby improving the representativeness and expressiveness of AIS data;

(3) Conducted validation and comparative simulation experiments, which show that the paths planned by the HPRM have higher safety and planning efficiency than those of conventional algorithms.

## 2. Data Preprocessing

The study uses a section of U.S. waters as an example, with AIS data sourced from [https://www.noaa.gov, accessed on 10 March 2024].

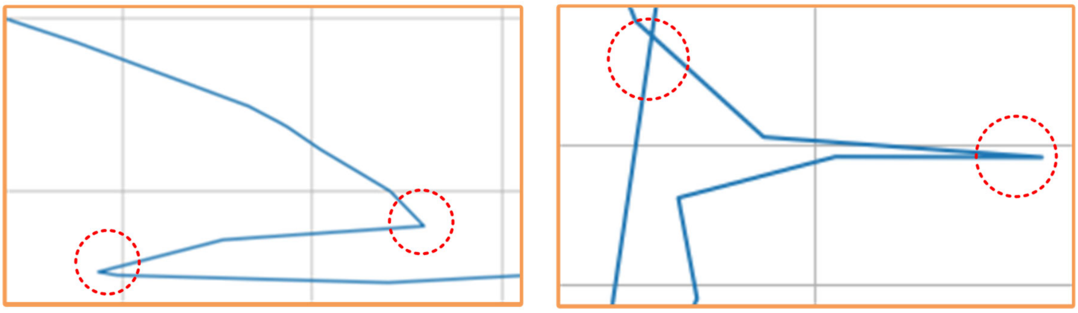
### 2.1. AIS Date

AIS (automatic identification system) is an automatic tracking system used for monitoring vessels and maritime traffic. It transmits both dynamic and static information of ships, including position, speed, heading, and vessel type, through VHF radio frequencies. AIS data are widely utilized in ship navigation, maritime monitoring, search and rescue operations, and environmental protection, contributing to enhanced maritime safety and traffic management efficiency. AIS data are collected via terrestrial receiving stations and satellites, and after compression and analysis, they are used to predict navigation trajectories, detect collision risks, and identify anomalous behaviors. In marine transportation and research, AIS data play a crucial role.

By thoroughly analyzing historical AIS data, it is possible not only to better adapt to the current marine environment but also to learn from past navigation tracks, thereby avoiding the repetition of previous navigation errors. Traditional path planning algorithms often overlook specific geographical features, such as islands, land locations, and underwater reefs and shoals near ports. These obstacles are often difficult to clearly identify on maps.

### 2.2. Data Cleaning

In the data processing phase of this study, we conducted meticulous screening and cleaning to ensure the accuracy of the analysis. First, we excluded invalid data that did not align with the research objectives, specifically vessel records that were in non-navigational states (such as anchoring, mooring, or grounding). Figure 3 is a schematic diagram of common erroneous AIS data, indicating that this study needs to delete the erroneous data.



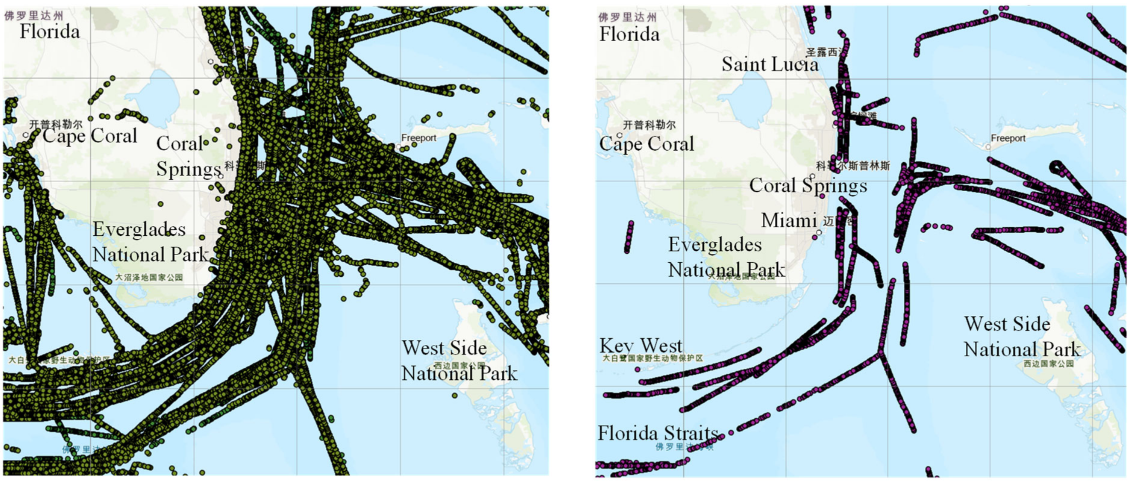
(a) (b)

**Figure 3.** Cleaning of erroneous data. (**a**) Type I: sharp turning; (**b**) Type II: self crossing.

Next, we identified and removed erroneous data that did not adhere to parameter specifications, including records with missing key information, unreasonable latitude and longitude positions, abnormal speeds, or courses exceeding 360 degrees. These errors typically originate from data packet losses in the physical layer communication links.

Additionally, we addressed duplicate data by categorizing them into two types. For data entries that were completely identical, we retained only one record to avoid redundancy. For data packets with the same MMSI (maritime mobile service identity) and timestamp but different data parameters, we adopted a more cautious approach. Since the authenticity of these data packets could not be determined, we chose to delete them to prevent potential interference and confusion. Through these steps, we ensured the purity of the research dataset and the reliability of the analysis results.

When processing automatic identification system (AIS) data, we often face the challenge of managing vast amounts of information. To ensure the specificity and effectiveness of the analysis, it is essential to meticulously filter the data to extract information closely related to the analytical objectives. Figure 4 is a comparison of AIS data before and after cleaning. This process involves multi-dimensional considerations, including but not limited to geographic region, time range, vessel type, and navigational status.




(a) (b)

**Figure 4.** Comparison of AIS data before and after cleaning. (**a**) Raw AIS data; (**b**) AIS data after cleaning.

In order to facilitate discussion, the data cleaning part will be completed using the filter function of Excel software. It is generally recognized that the draft of medium-sized container ships is usually around 10 m to 12 m. Therefore, we will filter out the data of ships with a draft of more than 10 m. At the same time, we can filter out ships with SOG greater than 2, COG between 0° and 360°, and longitude and latitude should also be between reasonable areas.

## 3. Improved DP Compression Algorithm

In this paper, we introduce an improved DP algorithm specifically designed for the efficient compression of vessel trajectory data. Given the massive scale of AIS data, especially in busy shipping lanes and port areas, this efficient compression algorithm is crucial as it can significantly reduce storage requirements, accelerate data transmission speeds, and enhance data processing efficiency.

### *3.1. Algorithm Principle*

The DP compression algorithm identifies key points in a curve through a recursive approach: first, it forms a line segment between the starting point and the ending point and finds the point on the curve that deviates the most from this line segment. If the deviation distance is less than the threshold, the intermediate points are approximated with a straight line; otherwise, the point with the maximum deviation is designated as a new key point, and the process is recursively applied. This continues until the error is within the threshold range, thereby simplifying the curve while retaining its main features.

This set of figures illustrates the basic concept of DP compression: within a series of state nodes, a maximum allowable distance $d_{max}$ is set. When $d > d_{max}$, the corresponding point is retained; and when $d < d_{max}$, the point is removed, thereby compressing the path. By progressively eliminating unnecessary intermediate nodes and retaining only the key states, the path is simplified. In Figure 5, this process is clearly illustrated. This method is widely used in path optimization, compression, and data simplification. However, for AIS data, simple parameter compression alone does not sufficiently meet practical requirements, as it is necessary to preserve key nodes. Therefore, we introduce multiple threshold parameters to retain the critical points of the vessels.

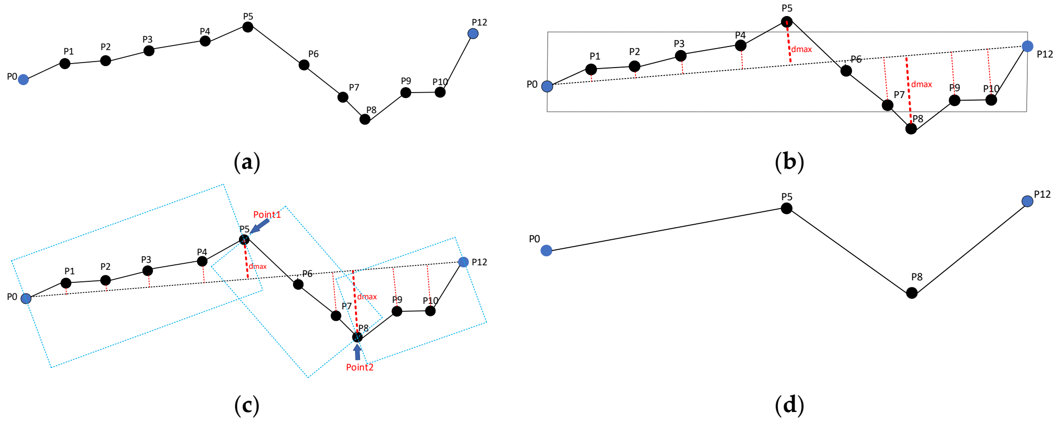



**Figure 5.** Principle of the DP algorithm. (**a**) Initial route segment; (**b**) Determine whether the point needs to be retained based on the set threshold; (**c**) Select key points based on the threshold, divide the path into two segments, and repeat this operation until all key points are selected; (**d**) Finally, only the key points are retained, and the rest are discarded to form a simplified path while retaining the main shape features of the original path.

However, the DP algorithm utilizes planar Cartesian coordinate data, whereas ship trajectory data (typically based on geographic coordinate systems) involve calculating spherical distances. For our purposes, it is challenging to directly compute the distance between trajectory points and lines using geographic coordinates. Therefore, to facilitate the calculation of point-to-line distances, we can convert geographic coordinates to Mercator projection coordinates. In Equation (1), $(\lambda, \varphi)$ represents the longitude and latitude of a trajectory point. $\varphi_0$ denotes the standard latitude in the Mercator projection, and $e$ represents the first eccentricity of the Earth's ellipsoid. The Mercator projection coordinates $(X, Y)$ of a trajectory point are calculated as follows:

$$\begin{gathered} X = \lambda \frac{a \cos\varphi_0}{\sqrt{1-e^2 \sin^2 \varphi_0}} \\ Y = \frac{a \cos\varphi_0}{\sqrt{1-e^2 \sin^2 \varphi_0}} \ln\left( \tan\left(\frac{\pi}{4} + \frac{\varphi}{2}\right) \left(\frac{1-e\sin\varphi}{1+e\sin\varphi}\right)^{\frac{e}{2}} \right) \end{gathered} \tag{1}$$

### 3.2. Threshold Design

The traditional Douglas–Peucker (DP) algorithm primarily focuses on geometric attributes but neglects the dynamic environmental changes during navigation, such as ocean currents, wind speed, and the movement of other vessels. These dynamic factors are crucial in actual maritime navigation and can significantly impact the accuracy of vessel trajectories. Additionally, during the trajectory simplification process, key turning points or locations near important navigational markers may be inadvertently removed, thereby reducing the practicality and safety of the routes.

To address these issues, we have implemented the following key optimizations to the DP algorithm:

Consideration of Speed and Heading Changes: By setting thresholds for speed and heading changes, the improved algorithm retains key points that exhibit significant variations in speed or heading. This measure ensures that the dynamic characteristics of the navigation data are preserved, significantly enhancing the precision and safety of path planning.

Dynamic Segmentation: The new algorithm segments the trajectory based on high curvature or significant distance differences to better adapt to complex navigational environments. This approach is particularly effective in handling abrupt trajectory changes caused by environmental variations or vessel maneuvers.

Data Compression and Quality Assurance: By repeatedly applying the improved standards, the algorithm not only effectively compresses the data but also ensures data quality by precisely selecting which data points to retain.

The calculation of thresholds plays a crucial role in compressing vessel trajectory paths. For AIS data, a higher threshold implies a higher compression rate, resulting in fewer latitude and longitude points being retained. However, this also means that fewer relevant features of the vessel trajectories are preserved. Therefore, it is essential to set appropriate thresholds to compress the paths effectively. In the improved DP algorithm, we consider four key vessel features, which are as follows:

(1) Geometric threshold determination: Calculate the perpendicular distance $D_i$ each intermediate point $(P_i(x_i, y_i)$ in a sequence relative to the line defined by the start and end points $P_1(x_1, y_1)$ and $P_N(x_N, y_N)$. For each point, this distance is calculated using the following formula:

$$D_i = \frac{|(y_N - y_1)x_i - (x_N - x_1)y_i + (x_N \cdot y_1 - x_1 \cdot y_N)|}{\sqrt{(y_N - y_1)^2 + (x_N - x_1)^2}} \tag{2}$$

If $(D_i)$ exceeds a preset geometric threshold, the segment is considered to require preservation in order to capture more geometric details of the path.

(2) Velocity difference threshold determination: For each pair of consecutive points $P_i$ and $P_{i+1}$, calculate the difference in velocity:

$$\Delta V_i = |V_{i+1} - V_i| \tag{3}$$

If $\max(\Delta V_i)$ exceeds the set velocity threshold, it indicates significant velocity changes in this path segment, which may affect navigational stability. Therefore, more points should be retained to capture detailed variations in velocity.

(3) Heading change threshold determination: Calculate the heading change between consecutive points:

$$\Delta H_i = |\text{mod}(H_{i+1} - H_i + 180, 360) - 180| \quad (4)$$

where $H_i$ and $H_{i+1}$ are the headings of points $P_i$ and $P_{i+1}$, respectively. If $\max(\Delta H_i)$ exceeds the preset heading threshold, it indicates a significant change in heading within this path segment. To ensure the completeness and usefulness of navigation data, these key points should be retained.

(4) The curvature is defined as follows: as $M'$ approaches M along curve L, if the limit of the mean curvature of arc $\overline{MM'}$ exists, this limit is called the curvature of curve $L$ at point $M$, denoted as $K$.

$$K = \lim_{\Delta s \to 0} \left| \frac{\Delta \alpha}{\Delta s} \right| = \left| \frac{d\alpha}{ds} \right| = \frac{f(x)''}{\left(1 + f(x)'^2\right)^{\frac{3}{2}}} \quad (5)$$

The equation $y = f(x)$ for the path is inconvenient to solve, so we transform this formula to derive the calculation formula for the curvature threshold.

$$k_i = \frac{4 \cdot \sqrt{s_i(s_i - a_i)(s_i - b_i)(s_i - c_i)}}{a_i \cdot b_i \cdot c_i} \quad (6)$$

Here, $s_i = \frac{a_i + b_i + c_i}{2}$, where $a_i,\ b_i\ , c_i$ are the side lengths of the triangle formed by three consecutive points $P_{i-1},\ P_i$, and $P_{i+1}$. If $\max(k_i)$ exceeds the set curvature threshold, it indicates that this segment of the path contains a significant curve or turn. To prevent loss of path information, this segment should be retained.

By defining these thresholds, we ensure that some critical points are retained when using the DP (Douglas–Peucker) algorithm for path simplification.

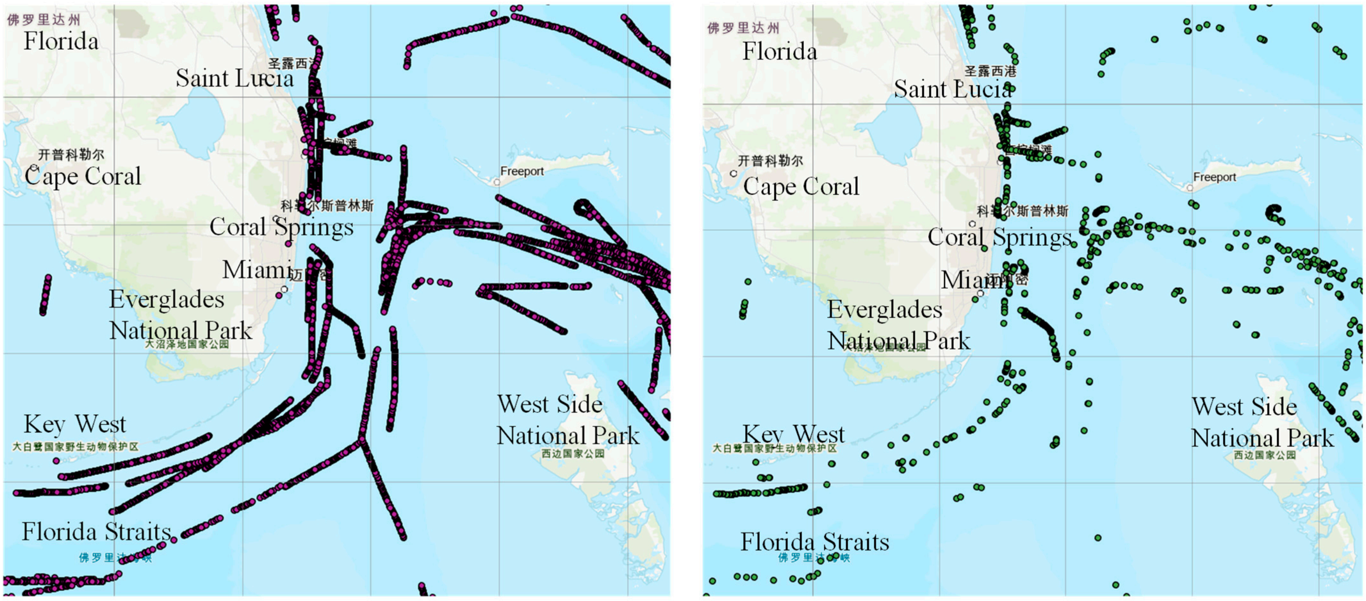
The pseudocode of Algorithm 1 is as follows:

**Algorithm 1** Improved DP Algorithm with Multiple Thresholds

**Require:** Trajectory points $P = \{P1,\ P2,\ \ldots Pn\}$, thresholds $T_S,\ T_v, T_h,\ T_c$
**Ensure:** Simplified trajectory points $Q$
1: $Q \leftarrow \{P1,\ Pn\}$
2: Initialize stack $S \leftarrow [(P1,\ Pn)]$
3: **while** S is not empty **do**
4: $(P_{start},\ P_{end}) \leftarrow$ S.pop()
5: $d_{max},\ i_{max} \leftarrow 0, 0$
6: **for** $i\ \leftarrow\ index(P_{start}) + 1\ index(\ P_{end}\ )\ - 1$ **do**
7: Calculate $D_i, \Delta v_i, \Delta H_i,\ k_i$
8: **if** any threshold exceeded **then**
9: $d_{max} \leftarrow$ value exceeding threshold
10: $i_{max} \leftarrow i$
11: **end if**
12: **end for**
13: **if** $d_{max} >\ T_S$ **then**
14: Insert $P_{\,i_{max}}$ into Q
15: S.push(($P_{start}, P_{\,i_{max}}$))
16: S.push(($P_{\,i_{max}}, P_{end}$))
17: **end if**
18: **end while**
19: **return** $Q$

This algorithm simplifies the data points in the compressed ship path through iteration, retaining those points that have a significant impact on the path's shape, speed, heading, and curvature.

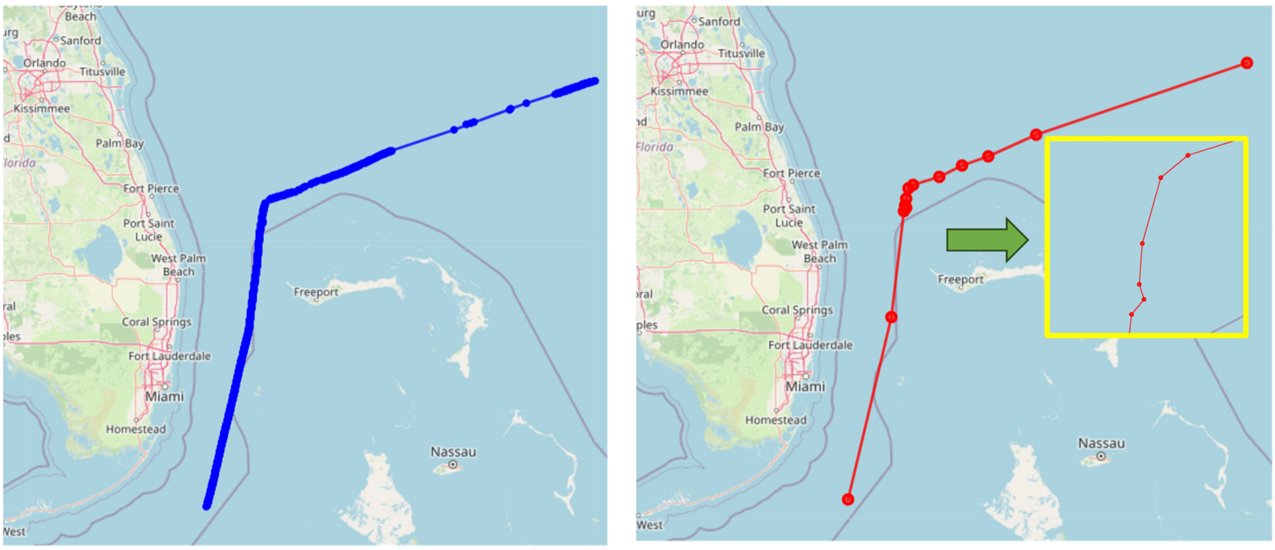
In the Figure 6, the left side displays the vessel AIS data for 1 January 2020, while the right side shows the data after compression using the improved DP algorithm. The results demonstrate that the improved DP algorithm can significantly compress AIS data, achieving a compression rate of over 80%.




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

**Figure 6.** Comparison of AIS data of some coastlines on a certain day before and after the improved DP algorithm compression. (**a**) AIS data after cleaning; (**b**) AIS data after compression using the improved DP algorithm.

Figure 7 is about compressing the route of the same vessel. The results in the figure indicate that the improved DP algorithm significantly enhances the compression rate of AIS data in comparative experiments and also performs excellently in retaining key points.




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

**Figure 7.** Comparison of a single route before compression and after using the improved DP algorithm. (**a**) Initial route of the ship; (**b**) The route after compression using the improved DP algorithm.

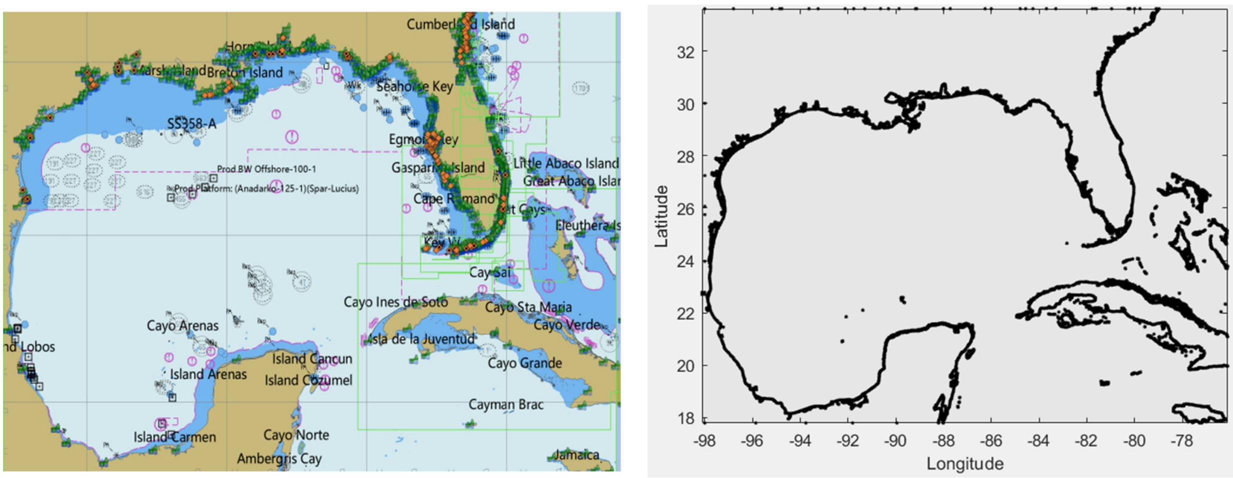
# 4. Hybrid Probabilistic Road Map (HPRM)

## 4.1. Environment Map Import

In traditional path planning, grid maps are a commonly used method for environmental representation. Grid maps divide the environment into discrete cells or pixels, where each cell's navigability is indicated in binary form. Although this approach is relatively simple to implement and can meet the basic needs of path planning, it suffers from limited resolution, leading to poor accuracy, particularly when dealing with areas characterized by complex terrain or dynamic maritime environments. Furthermore, the spatial representation capability of grid maps is constrained, making it difficult to accurately depict detailed geographic features. As a result, they exhibit significant limitations when processing complex geographical environments, such as coastlines and island distributions.

To address these issues, this study delves into the analysis of S-57 nautical chart files, extracting key geographic information contained within, especially the latitude and longitude data and other maritime features from the Gulf of Mexico and the Caribbean region. As an internationally recognized electronic chart standard, S-57 charts provide rich geographic and navigational data, including water depth, shipping lanes, obstacles, and navigational markers [33,34].

Figure 8 presents the environmental map construction results for the Gulf of Mexico and parts of the Caribbean Sea. The map not only accurately delineates the regional boundaries but also achieves a higher level of precision in displaying details, capturing the complex terrain and key navigational features within the area. With this environmental map, the study is able to better simulate real-world navigation scenarios, thereby enhancing the adaptability and reliability of the path planning process in actual environments.




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

**Figure 8.** Environment map construction. (**a**) Electronic navigational charts; (**b**) Environmental maps created by analyzing nautical chart data.

## 4.2. Selection of Sampling Points

The traditional PRM algorithm uses random sampling, but this often results in many invalid sampling points, such as those located in obstacles or non-navigable areas. These points cannot be used for path planning, thus wasting computational resources. Especially in complex environments, sparse sampling may cause some critical navigable paths to be overlooked.

AIS points represent the actual navigation paths and positions of vessels, reflecting areas with high navigation density in the waterway. These data typically show commonly used routes and real navigation habits for avoiding hazardous areas, enabling the generation of routes that better meet practical needs. However, the large volume of AIS data can reduce algorithm efficiency if used directly. Section 3 introduces an algorithm for compressing AIS data, and the resulting data are then used as sampling points for the PRM algorithm, which can significantly enhance the rationality and practicality of path planning.

Since intersections can occur between different routes, we can calculate these intersections to identify navigable points along the path. A vessel's historical route consists of numerous latitude and longitude points. We treat each pair of adjacent latitude and longitude points as a short route segment, so each segment consists of only two points. Then, by setting up equations for all route segments, we can calculate the intersections between different routes.

Assume route segment AB is defined by points $A(x_1,\ y_1)$ and $B(x_2,\ y_2)$, and route segment CD is defined by points $C(x_3,\ y_3)$ and $D(x_4,\ y_4)$.

Solve the system of equations to find the intersection point as follows:

$$\begin{bmatrix} x_2 - x_1 & -(x_4 - x_3) \\ y_2 - y_1 & -(y_4 - y_3) \end{bmatrix} \begin{bmatrix} t \\ s \end{bmatrix} = \begin{bmatrix} x_3 - x_1 \\ y_3 - y_1 \end{bmatrix} \tag{7}$$

Let the matrix be $A$ and the vector be $b$, then we have the following:

$$A \begin{bmatrix} t \\ s \end{bmatrix} = \mathrm{b} \tag{8}$$

The matrix $A$ can be represented as follows:

$$A = \begin{bmatrix} x_2 - x_1 & -(x_4 - x_3) \\ y_2 - y_1 & -(y_4 - y_3) \end{bmatrix} \tag{9}$$

The vector $b$ is as follows:

$$b = \begin{bmatrix} x_3 - x_1 \\ y_3 - y_1 \end{bmatrix} \tag{10}$$

To solve for $t$ and $s$, use the inverse of the matrix.

$$\begin{bmatrix} t \\ s \end{bmatrix} = A^{-1} b \tag{11}$$

The inverse of matrix $A$ (if it exists) is as follows:

$$A^{-1} = \frac{1}{(x_2 - x_1)(y_4 - y_3) - (y_2 - y_1)(x_4 - x_3)} \begin{bmatrix} -(y_4 - y_3) & x_4 - x_3 \\ -(y_2 - y_1) & x_2 - x_1 \end{bmatrix} \tag{12}$$

If the denominator is 0, it indicates that the two line segments are either parallel or collinear, and there is no unique intersection point.

Determine the intersection point.

After solving for the parameters $t$ and $s$, substitute $t$ into the parametric equation of line segment $AB$, or $s$ into the parametric equation of line segment $CD$ to obtain the coordinates of the intersection point: $x = x_1 + t(x_2 - x_1), y = y_1 + t(y_2 - y_1)$;

This intersection point is the intersection of the two line segments. If $0 \le t \le 1$ and $0 \le s \le 1$, then the intersection point lies within the bounds of both segments. Otherwise, the intersection point lies on the extension of the segments, outside the actual line segments themselves.

Calculate the intersection points between AIS trajectory segments, and use these intersection points along with the compressed AIS data points as fundamental sampling points. Figure 9 is about the intersection points between the routes. These points can be considered as navigable points in the PRM algorithm, serving as a foundation for preliminary route planning.

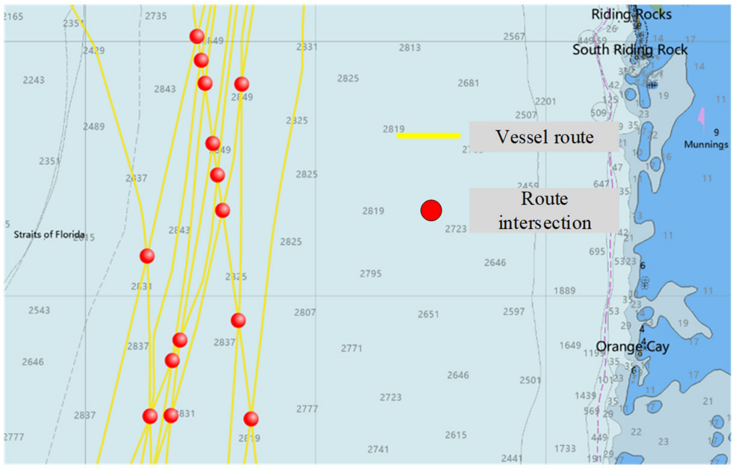



**Figure 9.** Intersection diagram between routes.

### *4.3. HPRM Algorithm Design*

#### 4.3.1. Constructing Path Segments Between Nodes

When constructing connecting path segments, the Haversine formula is used to calculate the spherical distance between nodes, and a distance matrix is generated. Each node selects several of its nearest neighboring nodes for connection based on the distance matrix, thereby forming a sparse graph. To avoid unreasonable long-distance connections, a maximum allowable connection distance can be set, and only neighboring nodes within this distance range are considered for connections. This method ensures the rationality of connections, making the generated paths more consistent with the actual geographical conditions while also ensuring computational efficiency. The Haversine formula is as follows:

$$d = 2r\arcsin\left(\sqrt{\sin^2\left(\frac{\phi_2 - \phi_1}{2}\right) + \cos{(\phi_1)}\cos{(\phi_2)}\sin^2\left(\frac{\lambda_2 - \lambda_1}{2}\right)}\right) \quad (13)$$

In the Haversine formula, $\varphi$ represents latitude, $\lambda$ represents longitude, R is the Earth's radius (6371 km), and $d$ is the spherical distance between two points.

#### 4.3.2. Collision Detection Between Path Segments

The collision detection step employs a grid mapping detection method for obstacle identification. During map construction, the area is processed into a grid, with each grid cell indicating whether it represents an obstacle region. When evaluating a candidate path segment, each grid cell that the path traverses is individually checked to determine if it contains an obstacle. If the path intersects an obstacle region, that edge is marked as invalid and is not included in the node's edge list. This grid-based collision detection accurately identifies obstacles along the path, ensuring that the generated path is feasible.

To enhance the precision of grid mapping detection, the size of the grid should be appropriately set. If the grid size is too large, the deviation of the planned path will significantly increase. Figure 10 is about inflating obstacles to enhance navigation safety. Conversely, if the grid size is too small, the resulting path may run too close to the coastline, which does not meet practical requirements. Therefore, it is necessary to establish a reasonable grid size range. Additionally, by performing simple obstacle grid inflation, safety margins can be increased, and the system can better accommodate obstacles of various shapes [35–37].

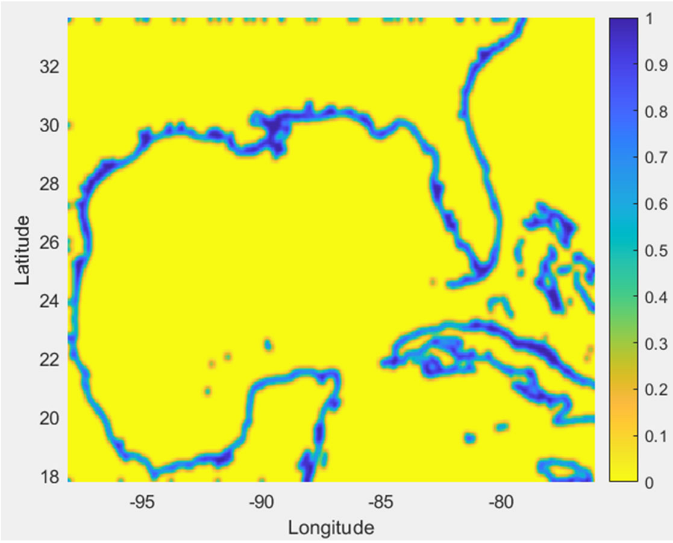



**Figure 10.** Inflate the drawn environment map with obstacles.

Figure 11 illustrates the relationship between nodes and their connecting edges, with green lines representing navigable segments of the path. The background gradient in the figure indicates varying risk levels across different regions, where blue areas denote high risk and yellow areas correspond to low risk. By employing an obstacle expansion technique, nodes and path edges are extended outward from high-risk regions, effectively reducing the potential for collisions. This approach enhances the safety of the path, particularly in complex environments, and significantly improves the robustness and navigational safety of the path planning process. The optimized nodes and edges in the figure not only cover low-risk areas but also ensure the connectivity of the path, thereby providing an efficient and safe navigation route for vessels [38].

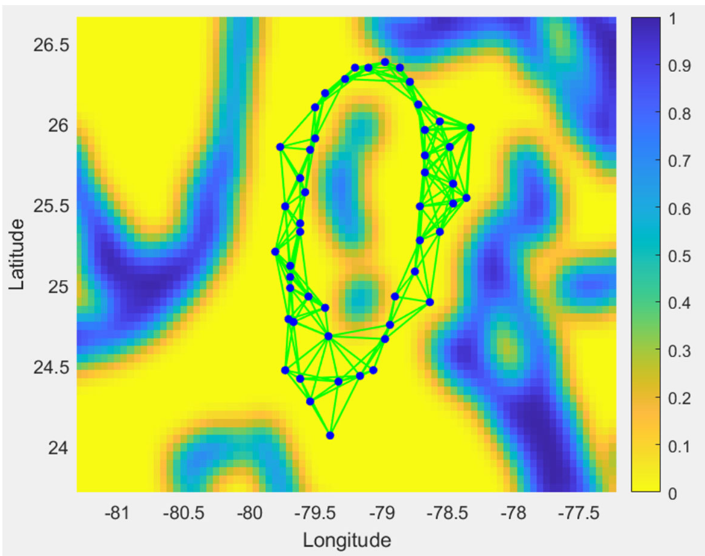



**Figure 11.** Routable routes generated by connecting nodes.

#### 4.3.3. Shortest Path Search

The graph structure is used to represent the relationships between nodes and their connecting edges, treating nodes as vertices in the graph and effective connecting edges as path segments. The Dijkstra algorithm is employed to search for the shortest path from the starting node to the target node within the graph. The Dijkstra algorithm can efficiently find the shortest path in weighted graphs with non-negative weights, making it one of the most commonly used single-source shortest path algorithms. The path search results provide a sequence of nodes from the start point to the end point, and these node sequences serve as the basis for further path optimization [39,40].

Figure 12 illustrates the path planning results generated using the HPRM algorithm in a complex maritime environment. The figure clearly marks the starting point (Start) and goal point (Goal), with a red line connecting these two points, representing the optimal navigation path computed by the algorithm. It is evident from the figure that the planned path intelligently avoids islands and other potential obstacles, ensuring navigational safety. Additionally, the path network, formed by blue nodes and green edges, explores multiple path options, ultimately generating a route that is both efficient and safe. This path planning approach demonstrates the effectiveness of the HPRM algorithm in avoiding obstacles within complex maritime areas, providing a theoretical foundation for ship navigation in real-world applications.

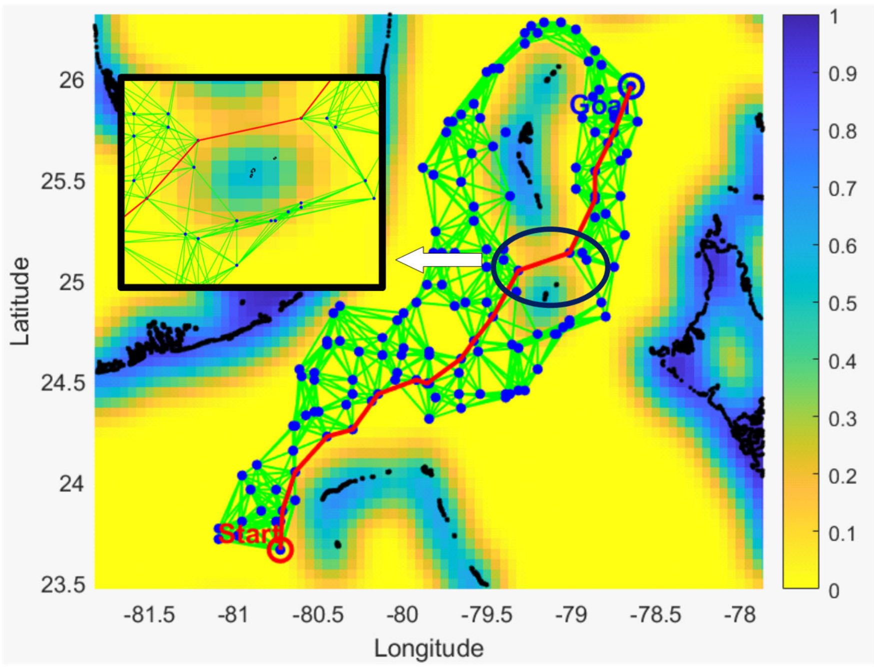



**Figure 12.** Path obtained by the HPRM.

*4.4. Basis Spline (B-Spline) Path Smoothing Optimization*

The B-spline expression can be defined as follows:

$$C(t) = \sum_{i=0}^{n} N_{i,p}(t) P_i \tag{14}$$

Here, $P_i$ represents the control points, $N_{i,p}(t)$ is the B-spline basis function, and $p$ is the degree of the curve.

The B-spline basis function $N_{i,p}(t)$ is defined recursively as follows:

When $p = 0$:

$$N_{i,0}(t) = \begin{cases} 1, & \text{if } t_i \leq t < t_{i+1} \\ 0, & \text{otherwise} \end{cases} \tag{15}$$

When $p > 0$:

$$N_{i,p}(t) = \frac{t - t_i}{t_{i+p} - t_i} N_{i,p-1}(t) + \frac{t_{i+p+1} - t}{t_{i+p+1} - t_{i+1}} N_{i+1,p-1}(t) \tag{16}$$

Here, $t_i$ represents the elements of the node vector, the objective function with a smoothing parameter.

In the optimization process of smoothing the B-spline, an objective function is used to balance the fitting accuracy and the smoothness of the curve:

$$J = (1 - p) \sum_{i=1}^{n} \parallel \mathrm{C}(t_i) - \mathrm{P}_i \parallel^2 + p \int_{t_0}^{t_f} \parallel \frac{d^2\mathrm{C}(t)}{dt^2} \parallel^2 dt \tag{17}$$

$(1 - p) \sum_{i=1}^{n} \parallel \mathrm{C}(t_i) - \mathrm{P}_i \parallel^2$ represents the fitting error of the curve to the data.

$p \int_{t_0}^{t_f} \parallel \frac{d^2\mathrm{C}(t)}{dt^2} \parallel^2 dt$ represents the smoothness of the curve.

By adjusting the smoothing parameter $p$, a trade-off can be made between the fitting accuracy and the smoothness of the curve.

Obstacle avoidance constraint: To ensure that the path avoids obstacles, the following constraints can be added: $d(\mathrm{C}(t), \mathrm{O}_j) \geq d_{\min}, \forall j, \forall t \in [t_0, t_f]$.

$d(\mathrm{C}(t), \mathrm{O}_j)$ represents the distance between the curve point $\mathrm{C}(t)$ and the obstacle $\mathrm{O}_j$, and $d_{\min}$ is the minimum safety distance. Figure 13 shows how the B-spline smooths and optimizes the path.

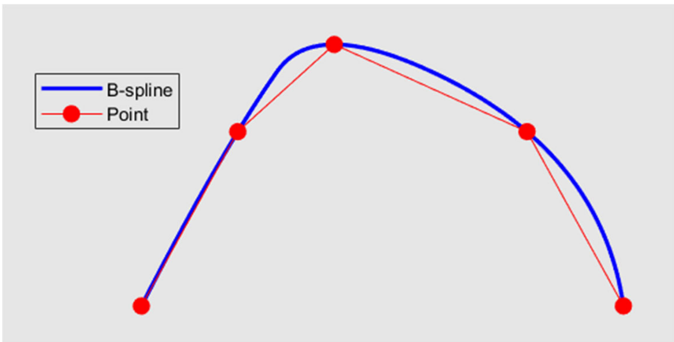



**Figure 13.** B-spline optimization principle.

Figure 14 shows that the use of B-spline curves can effectively optimize the smoothness of the path. Compared with the polyline path that directly connects the path nodes, the generated path is smoother and can eliminate sharp turns and discontinuous curvature changes [41,42].

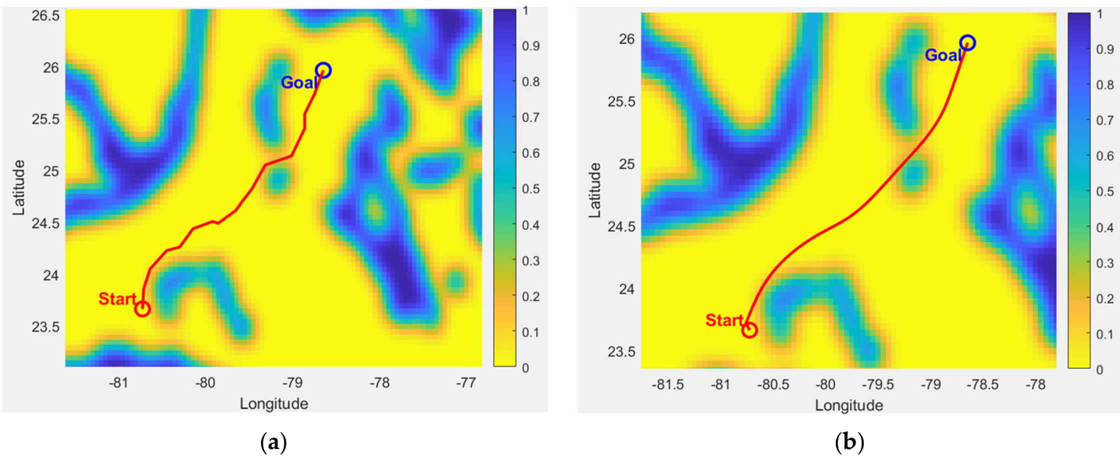



**Figure 14.** Path after B-spline smoothing. (**a**) The initial path obtained by planning; (**b**) The path after smoothing by the B-spline algorithm.

The following Algorithm 2 is the pseudocode of the HPRM algorithm:

**Algorithm 2** HPRM Algorithm

**Require:** Map, Start, Goal, n, k, Collision Check$(p1,\ p2)$
**Ensure:** A valid smoothed path from Start to Goal
1: $V \leftarrow \varnothing,\ E \leftarrow \varnothing$
2: while $|\ V\ | < n$ **do**
3: $p \leftarrow$ Random sample in map
4: **if** $p$ is navigable **then**
5: $V\ \leftarrow\ V \cup \{p\}$
6: **end if**
7: **end while**
8: **for all** $p\ \in\ V$ **do**
9: $N_P \leftarrow$ k-nearest neighbors of $p$
10: **for all** $p\prime \in\ Np$ do
11: **if** Collision Check$(p,\ p\prime)$ **then**
12: $E \leftarrow E\ \cup\ \{(p,\ p\prime)\}$
13: **end if**
14: **end for**
15: **end for**
16: Connect Start and Goal to $V$ by finding k-nearest neighbors
17: Perform graph search on $G = (V, E)$ to find path $P$
18: **for all** edges $(p,\ p\prime) \in P$ **do**
19: **if** not Collision Check$(p,\ p\prime)$ **then**
20: $E \leftarrow E \setminus \{(p,\ p\prime)\}$
21: Re-run path search
22: **end if**
23: **end for**
24: **Path Smoothing:** Apply B-spline smoothing on path P to generatesmoothed path Psmooth
25: **return** final smoothed path $P_{smooth}$ from Start to Goal

## 5. Experiment and Comparative Analysis

This section focuses on experimental validation of the improved DP algorithm and hybrid PRM, with a comparative analysis of their effectiveness and superiority. The study uses a section of U.S. waters as an example, with AIS data sourced from [https://www.noaa.gov, accessed on 5 May 2024].

### 5.1. Comparison of Different Compression Algorithms

To demonstrate the superiority of the improved DP algorithm, this study compared it with the k-means, RDP, and sliding window algorithms from multiple aspects, including compression rate comparison and the compression effects on the same trajectory.

From Figure 15, it can be analyzed that both the improved DP algorithm and the RDP algorithm are suitable for compressing AIS data. The k-means algorithm does not perform well in terms of compression, as it retains some unnecessary points and requires manual adjustment of clustering points for different vessels. Meanwhile, the sliding window algorithm transforms the compressed route into a straight-line path, resulting in excessively large turning radii at corners.

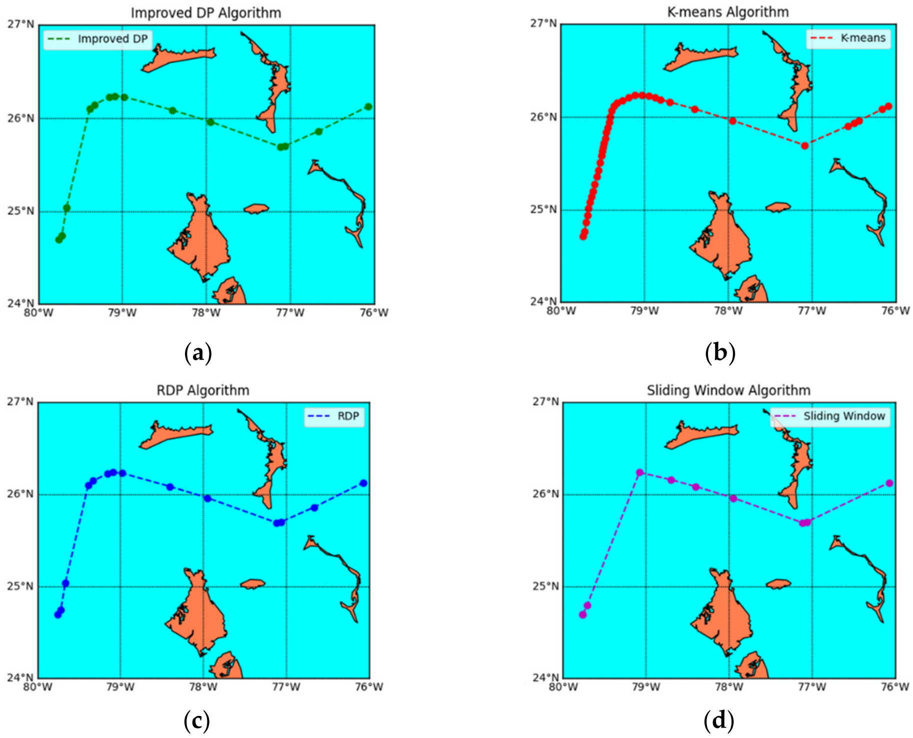



**Figure 15.** Comparison of different algorithms for compressing the same vessel data. (**a**) Improved DP algo-rithm; (**b**) k-means algorithm; (**c**) Ramer DP algorithm; (**d**) sliding window algorithm.

A key metric for evaluating a compression algorithm is its compression rate. Figure 16 clearly illustrates the comparison of four different compression algorithms across three different vessel IDs. Among them, the sliding window algorithm demonstrates the best performance in terms of compression rate, while the efficiency of the k-means algorithm is relatively low. Both the improved DP and RDP algorithms exhibit stable and robust performance. However, based on prior analysis, it is evident that the improved DP algorithm is particularly well-suited for compressing AIS data: it not only offers a high compression rate but also excels in retaining critical data points compared to other algorithms.

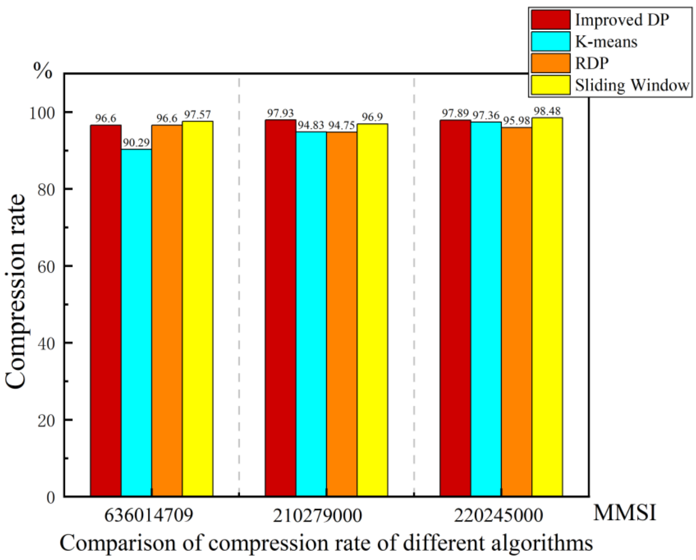



**Figure 16.** Compression rate comparison of different algorithms.

*5.2. Comparison of Path Planning Algorithms*

To verify the advanced nature of the proposed hybrid PRM algorithm, a specific geographic area near the waters of the United States was selected for testing. The latitude range is between 17.7915° and 33.6085°, and the longitude range is between −98.1167° and −76.1000°. This area covers parts of Florida, Georgia, and South Carolina, located mainly along the Atlantic coast. Within this region, random start and end points were chosen for path planning, and comparisons were made with the non-optimized PRM algorithm as well as the current mainstream algorithms, A* and Dijkstra. The results are as follows.

By combining Table 2 and Figure 17, it can be observed that, compared with the unoptimized PRM algorithm and the current mainstream algorithms A* and Dijkstra, the hybrid PRM algorithm has significant advantages in terms of planning time and route length. It generates the shortest processing time and shorter routes, making it suitable for applications that require rapid and efficient path planning. Additionally, as shown in the figure, the paths planned by the initial PRM algorithm have excessive turning points, insufficient curve smoothness, and are easily influenced by sampling points. In contrast, the A* and Dijkstra algorithms tend to become trapped in local optima, and the paths they generate do not meet the actual navigational requirements.

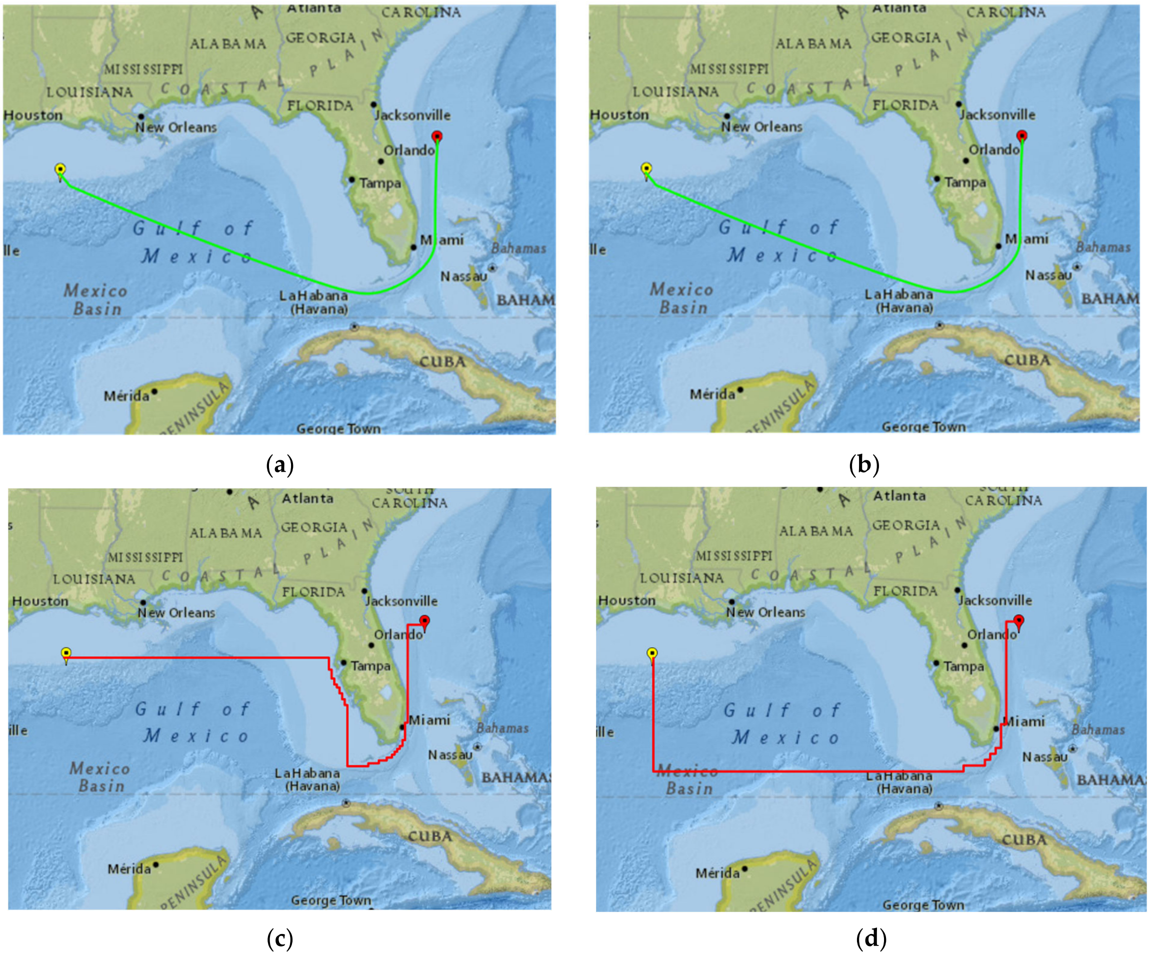



**Figure 17.** Path planning comparison diagram for different algorithms. (**a**) HPRM algorithm; (**b**) PRM algorithm; (**c**) A* algorithm; (**d**) Dijkstra algorithm.

**Table 2.** Comparison of path planning across different algorithms.

| Algorithm | Planning Times (s) | Route Length (nm) | Number of Turning Points |
|---|---|---|---|
| Hybrid PRM | 14.067670 | 1045.15 | 26 |
| PRM | 21.982659 | 1060.14 | 16 |
| A* | 28.236141 | 1239.69 | 36 |
| Dijkstra | 39.645747 | 1257.55 | 21 |

## 6. Conclusions

The hybrid PRM algorithm proposed in this study effectively addresses the limitations of the traditional PRM, with the following key innovations: (1) The application of an improved dynamic programming (DP) algorithm for compressing AIS data resolves the issue of excessive data volume, significantly enhancing the efficiency of subsequent path planning. (2) The sampling strategy of the PRM algorithm is improved by replacing random sampling points with compressed AIS latitude and longitude points, eliminating invalid samples, and simultaneously enhancing the safety of navigational planning. (3) B-splines are employed to optimize the smoothness of the path, addressing issues such as excessively large turning radii and unsmooth paths, thereby improving vessel maneuverability.

Comparative experiments demonstrate that the improved DP algorithm can effectively compress vessel trajectories, retain critical feature points, and extract valuable data. A comparison of different path planning algorithms highlights the efficiency and safety advantages of the proposed hybrid PRM algorithm. Experimental results show that the algorithm achieves an average reduction of approximately 10% in the length of the shortest paths generated, while efficiency is improved by over 35%.

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

**Funding:** This work was funded by the National Engineering Research Center of Ship & Shipping Control System and by the National Natural Science Foundation of China, grant number 52271322.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** All data are presented in the paper.

**Conflicts of Interest:** Author Bing Han was employed by the company Shanghai Ship and Shipping Research Institute Co., Ltd. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

## References

1. Wu, G.; Atilla, I.; Tahsin, T.; Terziev, M.; Wang, L. Long-voyage route planning method based on multi-scale visibility graph for autonomous ships. *Ocean Eng.* **2021**, *219*, 108242. [CrossRef]
2. Wu, G.; Li, H.; Mo, W. Improved Whale Optimization Algorithm for Maritime Autonomous Surface Ships Using Three Objectives Path Planning Based on Meteorological Data. *J. Mar. Sci. Eng.* **2024**, *12*, 1313. [CrossRef]
3. Svanberg, M.; Santén, V.; Hörteborn, A.; Holm, H.; Finnsgård, C. AIS in maritime research. *Mar. Policy* **2019**, *106*, 103520. [CrossRef]
4. Hart, E.; Timmis, J. Application areas of AIS: The past, the present and the future. *Appl. Soft Comput.* **2008**, *8*, 191–201. [CrossRef]

5. Robards, M.; Silber, G.; Adams, J.; Arroyo, J.; Lorenzini, D.; Schwehr, K.; Amos, J. Conservation science and policy applications of the marine vessel Automatic Identification System (AIS)—A review. *Bull. Mar. Sci.* **2016**, *92*, 75–103. [CrossRef]
6. 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]
7. Öztürk, Ü.; Akdağ, M.; Ayabakan, T. A review of path planning algorithms in maritime autonomous surface ships: Navigation safety perspective. *Ocean Eng.* **2022**, *251*, 111010. [CrossRef]
8. Song, Y.; Cao, X. Review of Intelligent Ship Path Planning Algorithms. *Front. Manag. Sci.* **2024**, *3*, 90–101. [CrossRef]
9. Wu, Y.; Wang, T.; Liu, S. A Review of Path Planning Methods for Marine Autonomous Surface Vehicles. *J. Mar. Sci. Eng.* **2024**, *12*, 833. [CrossRef]
10. Dijkstra, E.W. A note on two problems in connexion with graphs. In *Edsger Wybe Dijkstra: His Life, Work, and Legacy*; Association for Computing Machinery: New York, NY, USA, 2022; pp. 287–290.
11. Muhammed, M.L.; Humaidi, A.J.; Flaieh, E.H. Towards Comparison and Real Time Implementation of Path Planning Methods for 2R Planar Manipulator with Obstacles Avoidance. *Math. Model. Eng. Probl.* **2022**, *9*, 379–389. [CrossRef]
12. Laith, M.M.; Jaleel, H.A.; Hassan, F.E. A comparison study and real-time implementation of path planning of two arm planar manipulator based on graph search algorithms in obstacle environment. *Break. News Innov. Comput. Inf. Control.* **2023**, *17*, 61.
13. Muhammed, M.L.; Flaieh, E.H.; Humaidi, A.J. Embedded system design of path planning for planar manipulator based on chaos A* algorithm with known-obstacle environment. *J. Eng. Sci. Technol.* **2022**, *17*, 4047–4064.
14. Hart, P.E.; Nilsson, N.J.; Raphael, B. A formal basis for the heuristic determination of minimum cost paths. *IEEE Trans. Syst. Sci.* **1968**, *4*, 100–107. [CrossRef]
15. Kennedy, J. Swarm intelligence. In *Handbook of Nature-Inspired and Innovative Computing: Integrating Classical Models with Emerging Technologies*; Springer: Berlin/Heidelberg, Germany, 2006; pp. 187–219.
16. Jino Ramson, S.; Lova Raju, K.; Vishnu, S.; Anagnostopoulos, T. Nature inspired optimization techniques for image processing—A short review. In *Nature Inspired Optimization Techniques for Image Processing Applications*; Springer: Cham, Switzerland, 2019; pp. 113–145. [CrossRef]
17. Golberg, D.E. Genetic algorithms in search, optimization, and machine learning. *Addion Wesley* **1989**, *1989*, 36.
18. Kennedy, J.; Eberhart, R. Particle swarm optimization. In Proceedings of the ICNN'95-International Conference on Neural Networks, Perth, Australia, 6 August 2002; pp. 1942–1948.
19. Dorigo, M.; Gambardella, L.M. Ant colony system: A cooperative learning approach to the traveling salesman problem. *IEEE Trans. Evol. Comput.* **1997**, *1*, 53–66. [CrossRef]
20. Li, M.; Li, B.; Qi, Z.; Li, J.; Wu, J. Optimized APF-ACO algorithm for ship collision avoidance and path planning. *J. Mar. Sci. Eng.* **2023**, *11*, 1177. [CrossRef]
21. Elbanhawi, M.; Simic, M. Sampling-based robot motion planning: A review. *IEEE Access* **2014**, *2*, 56–77. [CrossRef]
22. Karaman, S.; Frazzoli, E. Sampling-based algorithms for optimal motion planning. *Int. J. Robot. Res.* **2011**, *30*, 846–894. [CrossRef]
23. LaValle, S.M.; Kuffner, J. Rapidly-exploring random trees: Progress and prospects. In *Algorithmic and Computational Robotics*; Taylor & Francis Group: New York, NY, USA, 2001; pp. 303–307. [CrossRef]
24. Kavraki, L.E.; Svestka, P.; Latombe, J.-C.; Overmars, M.H. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Trans. Robot. Autom.* **1996**, *12*, 566–580. [CrossRef]
25. Gu, Q.; Zhen, R.; Liu, J.; Li, C. An improved RRT algorithm based on prior AIS information and DP compression for ship path planning. *Ocean Eng.* **2023**, *279*, 114595. [CrossRef]
26. Zhen, R.; Jin, Y.; Hu, Q.; Shao, Z.; Nikitakos, N. Maritime anomaly detection within coastal waters based on vessel trajectory clustering and Naïve Bayes Classifier. *J. Navig.* **2017**, *70*, 648–670. [CrossRef]
27. Zhao, J.; Yan, Z.; Zhou, Z.; Chen, X.; Wu, B.; Wang, S. A ship trajectory prediction method based on GAT and LSTM. *Ocean Eng.* **2023**, *289*, 116159. [CrossRef]
28. Tang, C.; Wang, H.; Zhao, J.; Tang, Y.; Yan, H.; Xiao, Y. A method for compressing AIS trajectory data based on the adaptive-threshold Douglas-Peucker algorithm. *Ocean Eng.* **2021**, *232*, 109041. [CrossRef]
29. Zhao, L.; Shi, G. A trajectory clustering method based on Douglas-Peucker compression and density for marine traffic pattern recognition. *Ocean Eng.* **2019**, *172*, 456–467. [CrossRef]
30. Zhou, Z.; Zhang, Y.; Yuan, X.; Wang, H. Compressing AIS trajectory data based on the multi-objective peak Douglas–Peucker algorithm. *IEEE Access* **2023**, *11*, 6802–6821. [CrossRef]
31. Zhao, L.; Shi, G. A method for simplifying ship trajectory based on improved Douglas–Peucker algorithm. *Ocean Eng.* **2018**, *166*, 37–46. [CrossRef]
32. Marian-Vasile, S. Analysis of Submarine Operations in Semi-Enclosed Seas–The Black Sea. In *Proceedings of the 20th International Scientific Conference "Strategies XXI" Technologies—Military Applications, Simulation and Resources*; "Carol I" National Defence University: Bucharest, Romania, 2024; pp. 211–218.

33. Contarinis, S.; Nakos, B.; Tsoulos, L.; Palikaris, A. Web-based nautical charts automated compilation from open hydrospatial data. *J. Navig.* **2022**, *75*, 763–783. [CrossRef]
34. Clarke, J. A Proposed Submarine Electronic Chart Display and Information System. 2023. Available online: https://unbscholar.lib.unb.ca/items/c747099d-1c12-4710-af7c-2809fdac9b96/full (accessed on 15 January 2024).
35. Ayawli, B.B.K.; Appiah, A.Y.; Nti, I.K.; Kyeremeh, F.; Ayawli, E.I. Path planning for mobile robots using Morphological Dilation Voronoi Diagram Roadmap algorithm. *Sci. Afr.* **2021**, *12*, e00745. [CrossRef]
36. Yang, F.; Cao, C.; Zhu, H.; Oh, J.; Zhang, J. Far planner: Fast, attemptable route planner using dynamic visibility update. In Proceedings of the 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Kyoto, Japan, 23–27 October 2022; pp. 9–16.
37. Zhang, W.; Hou, Y.; Liu, H.; Lin, Z. Path planning for submersible surface ships in a three-dimensional environment considering safety distance. *J. Mar. Eng. Technol.* **2024**, 1–22. [CrossRef]
38. Yin, X.; Cai, P.; Zhao, K.; Zhang, Y.; Zhou, Q.; Yao, D. Dynamic path planning of AGV based on kinematical constraint A* algorithm and following DWA fusion algorithms. *Sensors* **2023**, *23*, 4102. [CrossRef]
39. Cho, S.W.; Park, H.J.; Lee, H.; Shim, D.H.; Kim, S.-Y. Coverage path planning for multiple unmanned aerial vehicles in maritime search and rescue operations. *Comput. Ind. Eng.* **2021**, *161*, 107612. [CrossRef]
40. Li, Q.; Ma, Q.; Weng, X. Dynamic path planning for mobile robots based on artificial potential field enhanced improved multiobjective snake optimization (APF-IMOSO). *J. Field Robot.* **2024**, *41*, 1843–1863. [CrossRef]
41. Liu, X.; Wang, X.; Wu, Z.; Zhang, D.; Liu, X. Extending Ball B-spline by B-spline. *Comput. Aided Geom. Des.* **2020**, *82*, 101926. [CrossRef]
42. Li, X.; Gao, X.; Zhang, W.; Hao, L. Smooth and collision-free trajectory generation in cluttered environments using cubic B-spline form. *Mech. Mach. Theory* **2022**, *169*, 104606. [CrossRef]