Journal of Marine Science and Application (2025) 24: 238-248
https://doi.org/10.1007/s11804-024-00430-5

RESEARCH ARTICLE

Ship Path Planning Based on Sparse A* Algorithm

Yongjian Zhai[1], Jianhui Cui[1], Fanbin Meng[2], Huawei Xie[2], Chunyan Hou[2] and Bin Li[2]

Received: 04 May 2023 / Accepted: 07 November 2023

Abstract

An improved version of the sparse A* algorithm is proposed to address the common issue of excessive expansion of nodes and failure to consider current ship status and parameters in traditional path planning algorithms. This algorithm considers factors such as initial position and orientation of the ship, safety range, and ship draft to determine the optimal obstacle-avoiding route from the current to the destination point for ship planning. A coordinate transformation algorithm is also applied to convert commonly used latitude and longitude coordinates of ship travel paths to easily utilized and analyzed Cartesian coordinates. The algorithm incorporates a hierarchical chart processing algorithm to handle multilayered chart data. Furthermore, the algorithm considers the impact of ship length on grid size and density when implementing chart gridification, adjusting the grid size and density accordingly based on ship length. Simulation results show that compared to traditional path planning algorithms, the sparse A* algorithm reduces the average number of path points by 25%, decreases the average maximum storage node number by 17%, and raises the average path turning angle by approximately 10°, effectively improving the safety of ship planning paths.

Keywords Sparse A* algorithm; Path planning; Rasterization; Coordinate transformation; Image preprocessing

1 Introduction

With the growing trend of economic globalization, trade exchanges between countries around the world have become increasingly close. In the face of rising transportation needs, the industry proposes high requirements for ship functions and safety in the new era. Especially in the context of the rapid development of modern science and technology, the industry promotes the development of ships in the direction of intelligence (Li et al., 2022c). One of the urgent problems for intelligent ships involves quick planning of an optimal route from the starting point to the destination and safely avoiding all obstacles in a short time when the ship is in a complex water area (i.e., a port) (Du et al., 2021). Therefore, several intelligent ship route planning algorithms have been developed.

Traditional global path planning algorithms, including the Dijkstra algorithm (Sedeno-Noda and Colebrook, 2019) and the A* algorithm (Grifoll et al., 2022; Hu et al., 2023; Cheng and Wang, 2023; Liu et al., 2023), are currently available. Sample-based path planning algorithms, including the probabilistic route map method (Zhou et al., 2020) and the fast search random tree algorithm (Qi et al., 2020), also exist. Furthermore, intelligent optimization global path planning algorithms, including the particle swarm optimization algorithm (Thi et al., 2017) and the genetic algorithm (Feng et al., 2022; Miao et al., 2021), are available. In addition, improvements to various algorithms and the combination of different algorithms (Li et al., 2022b) are observed. Among them, the A* algorithm has become one of the most widely used global path planning algorithms due to its search path stability (Phanthong et al., 2014; Li et al., 2017; Sang et al., 2021; Singh et al., 2018; Chen et al., 2022; Lee 2015; Miao et al., 2022). Wei et al. (2023) proposed an improved global path planning A* algorithm based on node optimization to expand the search domain of the A* algorithm and remove nodes by introducing random numbers, thus addressing the problem of the A* algorithm in

Article Highlights

- Improved the basic SAS algorithm from three aspects: gridding the nautical chart, constructing expandable areas, and adding constraint conditions.
- Implemented the FCM algorithm to perform hierarchical clustering on nautical charts, categorizing individual pixels based on their RGB values.
- Applied coordinate transformation algorithm to convert latitude and longitude coordinates into easily usable and analyzable Gaussian plane coordinates and grid coordinates.
- Accounted for the influences of vessel length, initial heading, and water depth on the outcomes of path planning.

✉ Jianhui Cui
jianhuicui@email.tjut.edu.cn

[1] Maritime College, Tianjin University of Technology, Tianjin 300384, China

[2] Tianjin Navigation Instrument Research Institute, Tianjin 300131, China

generating multiple inflection points and long paths. Ying and He (2022) introduced an improved A* algorithm that optimizes the grid map by setting a safe distance and presents the Floyd algorithm to reduce the number of path-turning points, thereby addressing the excessive closeness of the traditional A* algorithm to the dangerous area in the planned route.

Sparse A* search (SAS) is an improvement of the standard heuristic search algorithm A*, which effectively reduces the search space and enhances the efficiency and robustness of the algorithm by excluding invalid points and combining path constraints (Li et al., 2018; Zhou et al., 2021; Bi et al., 2019). Chen et al. (2012) added the random point method based on the sparse A* algorithm. The random function was used in the path planning area to uniformly increase the search nodes, and the visibility check method was employed to reduce the number of planned path points. However, the expansion of the algorithm will extend the algorithm space, and the maximum number of nodes consumed by the algorithm will significantly increase. Li and Liu (2017) combined the sparse A* algorithm with the cultural algorithm to propose a UAV dynamic path planning algorithm that can quickly avoid sudden threats and solve the path problem around the sparse A* algorithm. However, the algorithm disregards factors such as the initial heading, draft, and length of the ship. Thus, this algorithm in the planned path fails to address the practical application requirements of the ship.

Overall, with the emergence of numerous route-planning algorithms, route-planning technology has become increasingly mature. However, the limitations of traditional route planning algorithms have also become highly evident. For instance, in the course of chart processing and route planning, simultaneously considering the influence of the ship's initial heading, safety range, draft depth, and length on the planned route is impossible. Therefore, considering the aforementioned factors, this paper utilizes the sparse A* algorithm to plan an optimal obstacle avoidance route from the current to the target point, displaying high levels of safety and practicality.

Effectively reducing the average number of path points per unit path length and the average number of maximum storage nodes is possible using the ship route planning algorithm described in this paper. Additionally, considering factors such as the ship's initial heading, safety range, draft depth, and length, this algorithm enhances the safety and feasibility of the planned route.

2 Chart processing

2.1 Chart layering

In this paper, the image is layered in accordance with the chart color information. The target image is grayed after being inputted to reduce the difficulty of layering. However, grayscale images have 256 grayscale values. If only a few grayscale values are selected as the criteria for selecting information, then a large amount of information will be lost, resulting in a significant reduction in the security of the planned path. Therefore, this paper uses the fuzzy C-means clustering algorithm (FCM) to cluster gray values of the image and classify them within a certain threshold into the same type of information (Xie et al., 2018; Zhu, 2022).

FCM clustering and mean filtering are performed on the original nautical chart, and isolated points are removed. Shoal areas with water depths of 0–5, 5–10, 10–20, and over 20 m are selected as the effective layers of each chart.

2.2 Coordinate change

The position of the object in the chart is typically represented by the geographic coordinate system, namely the longitude and latitude coordinate system, and the longitude and latitude data are also used for ship positioning. The longitude and latitude coordinate systems can conveniently represent the position of any point on the Earth. However, the Earth is an irregular sphere with slightly flat poles and a slightly bulging equator. Thus, the longitude and latitude coordinate system is not conducive to the position representation on the map and the use of the algorithm. Considering the above problems, appropriate algorithms should be selected to convert longitude and latitude coordinates into Gaussian plane coordinates and grid coordinates that are easy to represent on the map.

In this paper, the Gauss–Kruger projection algorithm is first used to realize the mutual transformation of longitude and latitude coordinates and Gaussian plane coordinates (Chen, 2020). The Gauss–Kruger projection assumes that an elliptical cylinder is transversely sheathed outside the Earth's ellipsoid and is tangent to a meridian (this meridian is called the central or axis meridian). The central axis of the elliptical cylinder passes through the center of the ellipsoid and then uses a certain projection method to project the area within a certain range of longitude difference on both sides of the central meridian onto the elliptical cylinder and then expands this cylinder to become the projection plane. The Gauss–Kruger projection has no angular deformation, and its deformation in length and area is remarkably small. The central meridian has no deformation. From the central meridian to the edge of the projection zone, the deformation gradually increases. The maximum deformation is located at both ends of the equator in the projection zone. The Gauss–Kruger projection generally has the advantages of high projection accuracy, small deformation, and simple calculation (Li et al., 2022a). The Gauss–Kruger projection diagram is shown in Figure 1.

After converting longitude and latitude coordinates to Gaussian plane coordinates, Eq. (1) is then utilized to convert Gaussian plane coordinates to grid coordinates,

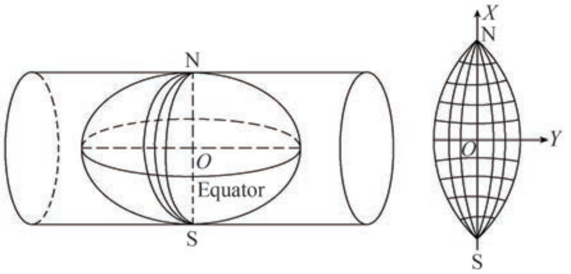


Figure 1 Gauss–Kruger projection diagram

while Eq. (2) is utilized to convert grid coordinates to Gaussian plane coordinates. The Gaussian plane coordinates obtained from the transformation of grid coordinates represent the Gaussian plane coordinates at the grid center point, where the grid coordinates are located.

$$\begin{cases} X' = \text{floor}\left(\dfrac{X - X_1}{\text{length}}\right) \\ Y' = \text{floor}\left(\dfrac{Y - Y_2}{\text{width}}\right) \end{cases} \tag{1}$$

$$\begin{cases} X = X_1 + X' \times \text{length} \\ Y = Y_2 + Y' \times \text{width} \end{cases} \tag{2}$$

where (X', Y') represent the grid coordinates of the point, (X, Y) represent the Gaussian plane coordinates of the point, (X_1, Y_1) represents the Gaussian plane coordinates of the upper left corner of the chart, (X_2, Y_2) represents the Gaussian plane coordinates of the lower right corner of the chart, length represents the actual length of each grid, and width represents the actual width of each grid.

2.3 Chart rasterization

The path planning algorithm must first construct the environment; that is, the appropriate method must be selected to describe the obstacle information. Such information is a crucial link in path planning and obstacle avoidance problems, which determines the advantages and disadvantages of the path. Common methods of environment construction include configuration space, free space, and grid methods. This paper uses the grid method to develop an environmental model. In the grid method, the environment is divided into a series of grids. Each grid is marked as a safe or an obstacle area according to the presence or absence of an obstacle. The granularity of the grid indicates the fineness of the environmental description. This method facilitates the creation and storage of environmental information and sets the foundation for the subsequent path-planning process.

In the grid map, the completely barrier-free grid corresponds to the accessible area, while the complete obstacle square corresponds to the unreachable area. Some obstacle grids are also regarded as inaccessible areas for safety reasons. If the unreachable grid is marked as a black color block and the reachable grid is marked as a white color block in the grid environment, then the environment information can be represented as Figure 2.

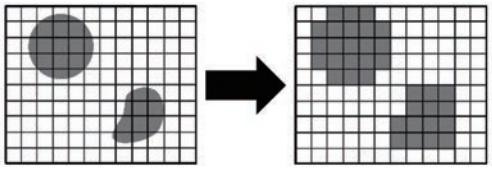


Figure 2 Obstacle grid

The traditional route planning algorithm uses a grid map with a specified length to reduce the versatility of the algorithm. However, some narrow areas that do not meet the requirements of the ship length are regarded as navigable areas due to an excessively small grid length, thus reducing the safety of the planned route. If the grid length is excessively long, then the entire grid with a small number of obstacle areas will be determined as an infeasible area, resulting in a reduction in the driving area and an increase in the planned path length. This paper considers the effect of ship length on grid size, adaptively adjusts grid size and density according to ship length, and grids the chart to address the aforementioned problems. The rasterization process is presented as follows:

1) The actual distance represented by the input chart is calculated as follows:

$$\begin{cases} \text{maplength} = X_2 - X_1 \\ \text{mapwidth} = Y_2 - Y_1 \end{cases} \tag{3}$$

where (X_1, Y_1) represent the Gaussian plane coordinates at the upper left corner of the chart, (X_2, Y_2) represent the Gaussian plane coordinates at the lower right corner of the chart, maplength represents the actual distance represented by the chart length, and mapwidth represents the actual distance represented by the chart width.

2) The actual distance represented by each pixel of the chart is calculated as follows:

$$\begin{cases} \text{Pixelslength} = \dfrac{\text{maplength}}{x\text{Pixels}} \\ \text{Pixelswidth} = \dfrac{\text{mapwidth}}{y\text{Pixels}} \end{cases} \tag{4}$$

where Pixelslength represents the actual length represented by each pixel on the chart, Pixelswidth represents the actual width represented by each pixel on the chart, *x*Pixels represents the number of horizontal pixels on the chart, and *y*Pixels represents the number of vertical pixels on the chart.

3) The number of pixels contained in each grid is calculated as follows:

$$\begin{cases} \text{horizontalpixel} = \text{ceil}\left(\dfrac{\text{shiplength}}{\text{Pixelslength}}\right) \\ \text{verticalpixel} = \text{ceil}\left(\dfrac{\text{shiplength}}{\text{Pixelswidth}}\right) \end{cases} \tag{5}$$

where horizontalpixel represents the number of pixels contained horizontally in each grid, verticalpixel represents the number of pixels contained horizontally in each grid, and shiplength represents the actual length of the ship.

4) The actual distance represented by each grid is calculated as follows:

$$\begin{cases} \text{Gridlength} = \text{horizontalpixel} \times \text{Pixelslength} \\ \text{Gridwidth} = \text{verticalpixel} \times \text{Pixelswidth} \end{cases} \tag{6}$$

where Gridlength and Gridwidth represent the actual length and width of each grid, respectively.

5) The number of rows and columns of the chart grid is calculated as follows:

$$\begin{cases} \text{maplines} = \text{ceil}\left(\dfrac{x\text{Pixels}}{\text{horizontalpixel}}\right) \\ \text{mapcolumns} = \text{ceil}\left(\dfrac{y\text{Pixels}}{\text{verticalpixel}}\right) \end{cases} \tag{7}$$

where maplines and mapcolumns represent the number of rows and columns of the chart grid, respectively.

6) According to the above data, the input chart is divided into maplines × mapcolumns grids, and the actual size of each grid is Gridlength × Gridwidth.

3 Sparse A* algorithm implementation

The sparse A* algorithm is an improvement of the traditional A* algorithm. This algorithm reduces the number of nodes and path points to be expanded and minimizes the search space by adding some constraints during node expansion (Wang et al., 2018; Li et al., 2022d).

The traditional A* algorithm fails to consider information such as the initial ship heading, ship length, draft depth, and other factors, yielding planned paths that do not meet the requirements of practical ship applications and have poor safety. The sparse A* algorithm has currently not been applied in the field of ship route planning. This paper comprehensively considers the aforementioned factors and uses the sparse A* algorithm to plan ship routes. The basic idea of this algorithm is similar to the traditional A* algorithm. First, the chart to be planned is imported and rasterized. Then, the cost to reach each node to be expanded is subsequently calculated through the preset cost function. The node to be expanded with the lowest cost is selected into the path point set, and its parent node is set as the current point. Finally, the extensible node is placed in the node set to be expanded. When the newly added node to the path point set is the target point, the source of the path point set is traced in accordance with the parent node until the starting point is identified and the path planning algorithm is obtained. The algorithm ultimately ends.

Compared with the traditional A* algorithm, the sparse A* algorithm only extends to some node elements in the neighborhood that meet the relevant constraints during node expansion. The region to be expanded at the next node of the algorithm comprises the current point S_i, the parent node $S_{i\text{-}1}$ of the current point, the expansion radius d, and the maximum expansion angle α decision. The identified area to be expanded is then divided into M parts, and the minimum cost point in each part is selected to be stored in the node set to be expanded. The node expansion method of the sparse A* algorithm is shown in Figure 3. The sector area in the figure is the area to be expanded.

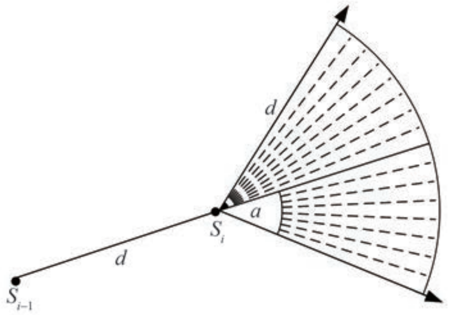


Figure 3 Areas to be expanded

3.1 Path planning process

The process of ship path planning based on the sparse A* algorithm is shown in Figure 4.

1) Input the chart and ship information, and then layer and rasterize the chart. Simultaneously, the longitude and latitude of the starting point are converted into the array coordinates available for the algorithm.

2) Create open and closed tables. The open table is used to record the nodes to be extended, and the closed table is used to record the extended nodes. Simultaneously, the starting point is inputted into the open table, and the closed table is left empty.

3) Determine whether the open table is empty. If it is empty, then the target path of the algorithm search failure is not found; if it is not empty, then S_i is inputted into the least expensive point to be extended in the open table into the closed table.

4) Identify whether S_i is the target point. If so, then execute step 5). The algorithm performs successful searches, and the target path is found. If not, then the extension node

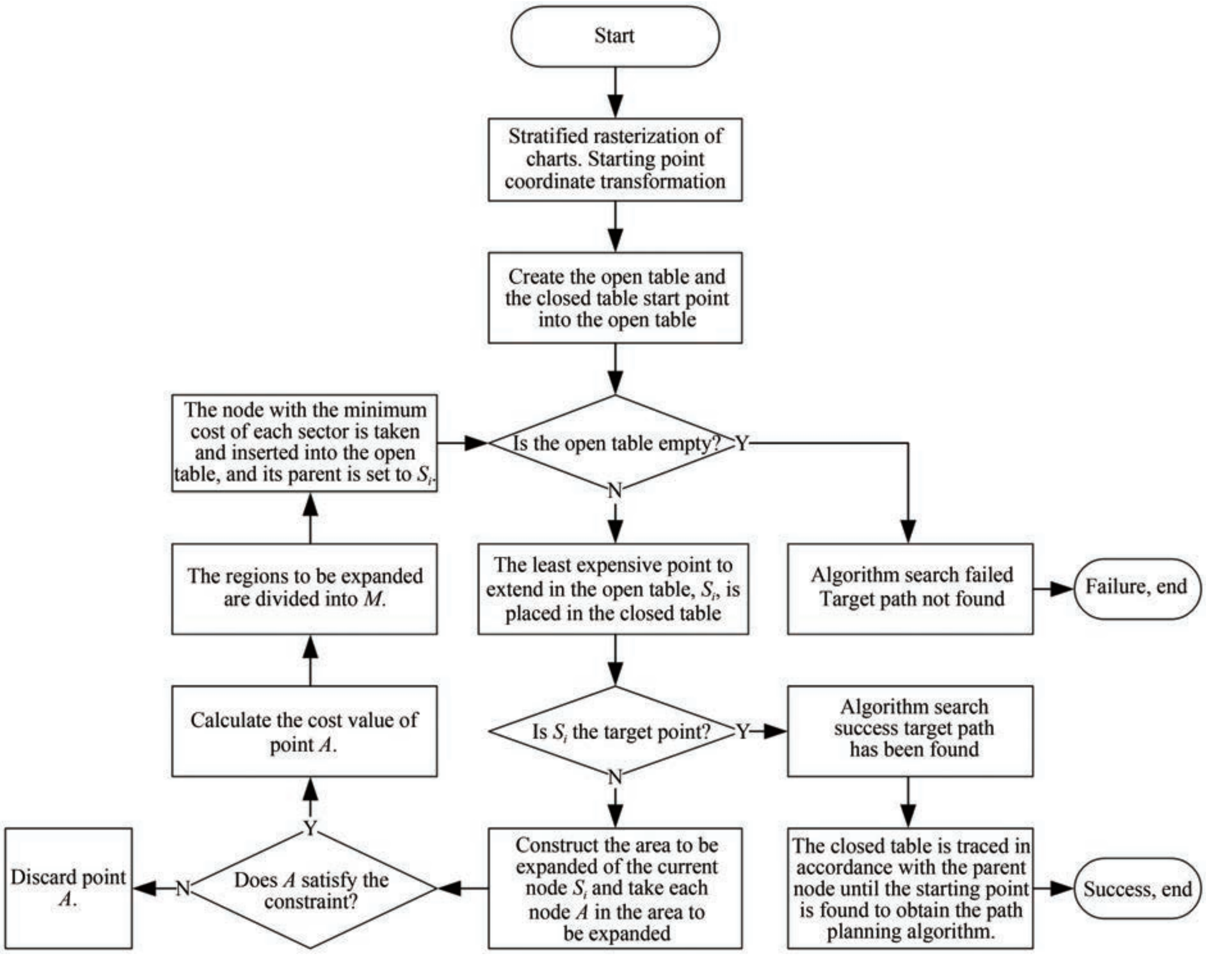

Start
Stratified rasterization of charts. Starting point coordinate transformation
Create the open table and the closed table start point into the open table
Is the open table empty?
Y
N
The node with the minimum cost of each sector is taken and inserted into the open table, and its parent is set to S_i.
The regions to be expanded are divided into M.
The least expensive point to extend in the open table, S_i, is placed in the closed table
Algorithm search failed Target path not found
Failure, end
Calculate the cost value of point A.
Is S_i the target point?
Y
N
Algorithm search success target path has been found
Y
Discard point A.
N
Does A satisfy the constraint?
Construct the area to be expanded of the current node S_i and take each node A in the area to be expanded
The closed table is traced in accordance with the parent node until the starting point is found to obtain the path planning algorithm.
Success, end

Figure 4 Flowchart of ship path planning based on the sparse A* algorithm

S_{i+1} of node S_i is obtained in accordance with the following four steps:

① Construct the area to be expanded for the current node S_i. The extension line of the connection between S_{i-1} and S_i, that is, the parent node of the current node, is taken as the axis of symmetry: twice the maximum expansion angle α is the angle, and the extension length d is the radius, forming a sector area as the area to be expanded for the current node S_i.

② Take grid node A in the area to be expanded to determine whether it meets the constraint. If not, then the point is discarded; if it is satisfied, then the generation value of this point is calculated as follows:

$$F(x) = g(x) + h(x) \tag{8}$$

where $F(x)$ represents the generation value of the current point, $g(x)$ represents the actual cost from the starting point to the current point, and $h(x)$ represents the estimated cost from the current point to the target point. The Euclidean distance is used in this paper to calculate $g(x)$ and $h(x)$, and the expression is:

$$\begin{cases} g(x) = \sqrt{\begin{aligned}&\left(\left|X'_{S_i} - X'_{S_{i+1}}\right| \times \text{length}\right)^2 \\ &+\left(\left|Y'_{S_i} - Y'_{S_{i+1}}\right| \times \text{width}\right)^2\end{aligned}} \\ h(x) = \sqrt{\begin{aligned}&\left(\left|X'_{S_i} - X'_{S_n}\right| \times \text{length}\right)^2 \\ &+\left(\left|Y'_{S_i} - Y'_{S_n}\right| \times \text{width}\right)^2\end{aligned}} \end{cases} \tag{9}$$

where (X'_{S_i}, Y'_{S_i}) represent the grid coordinates of the current point, (X'_{S_i+1}, Y'_{S_i+1}) represent the grid coordinates of the next extension point of the current point, (X'_n, Y'_n) represent the grid coordinates at the end point, length represents the actual length of each grid, and width represents the actual width of each grid.

③ Split the area to be expanded. The sector obtained from ① is divided into one sector every 5° to obtain M small sectors. A large M value yields a maximum expansion angle α. Identifying the track of the ship becomes difficult when the probability of finding the required track is high, and the required space for the algorithm will increase accordingly. Therefore, this paper sets the initial value of the expansion angle to 5° and gradually increases by 5° until the maximum expansion angle α is reached. On the premise of ensuring that the planned route can be identified, the feasibility of the planned route can be improved to the maximum extent, and the required space by the algorithm can be reduced.

④ Select the point to be expanded that meets the constraints and has the lowest generation value in each sector. This point is then stored in the open table of the node set to be expanded, and its parent node is set as the current node S_i.

5) Repeat steps 3) and 4). After finding the target path, trace the source of the closed table according to the parent node until the starting point is identified and the path planning algorithm is obtained.

3.2 Algorithm constraint settings

For a series of problems, such as the current state of the ship and some ship parameters that the traditional path planning algorithm fails to consider, the path planning constraints set in this paper mainly include the following:

1) Safety scope constraints

The ship is not a particle. When planning the path for the ship, the safe distance between the ship and the obstacle should be fully considered to ensure that the ship maintains a certain distance from the obstacle, reducing the possibility of ship grounding or reef accidents.

2) Draft constraint

Layered data from different water depth parts can be obtained after the chart is processed through the hierarchical grid process described in the first part. The navigable area can be obtained by importing the ship's draft and comparing it with the hierarchical chart data.

3) Initial bow restraint

The traditional path planning algorithm disregards the bow direction of the ship's initial position β. Therefore, the planned route is inconsistent with the actual application of the ship, which easily causes collision accidents during navigation. During ship path planning, the area to be expanded at the starting point is from the initial bow of the ship to the β decision. The area to be expanded at the starting point is shown in Figure 5.

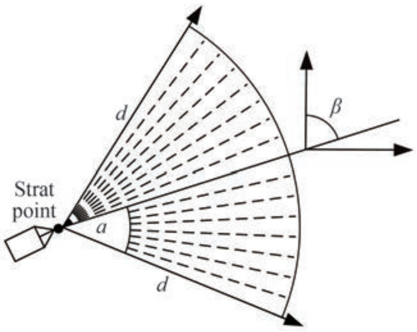


Figure 5 Area to be expanded at the starting point

4 Experiment comparison and result analysis

The experimental platform uses an Intel Core i5-6200U CPU@2.30 GHz processor and is programmed by Microsoft Visual Studio C++ for simulation experiments. The chart used in the experiment is selected from the areas of longitude E121°22'19"–E121°29'27" and latitude E39°45'32"–E39°42'54", as shown in Figure 6.

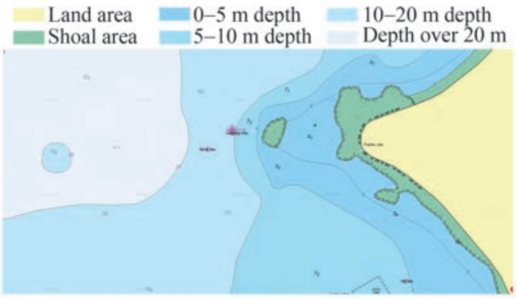


Figure 6 Experimental chart

The experimental vessel 1 is set with a length of 30 m, a safety range of 50 m, and a draft of 4 m. The starting point of the route is set a E121°26'36", N39°43'32", and the ending point of the route is set to E121°26'58", N39°45'27".

The experimental vessel 2 is set with a length of 50 m, a safety range of 80 m, and a draft of 8 m. The starting point of the route is set at E121° 25'37", N39° 43'52", and the ending point of the route is set to E121° 26'36", N39° 45'24".

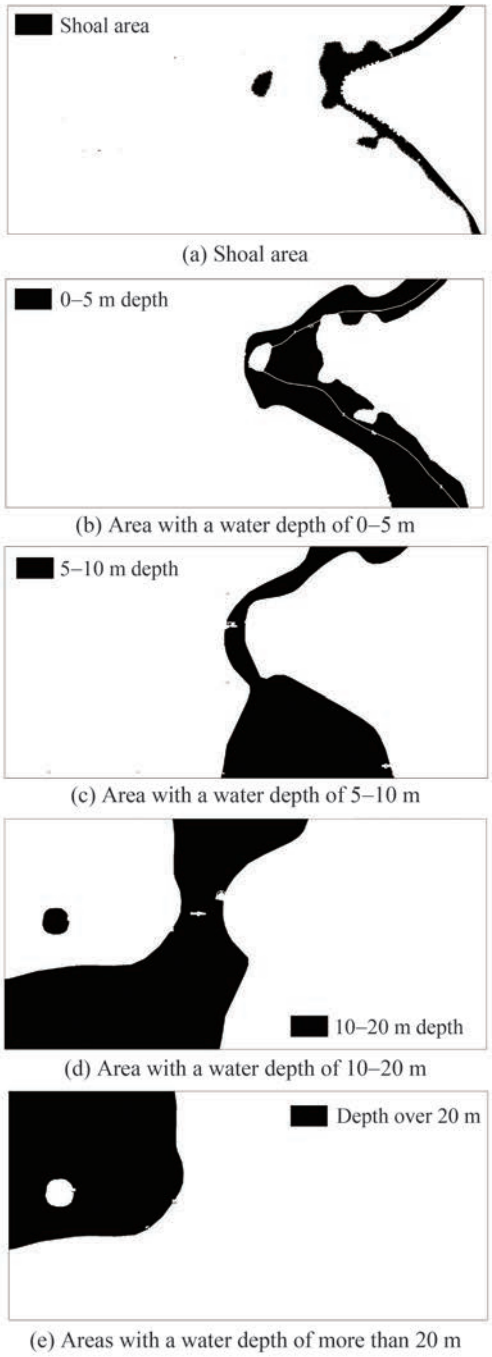
The bow directions of experimental vessels 1 and 2 are set at 0°, 90°, 180°, and 270°.

During the simulation experiments, ensuring that the safety range of the experimental vessel is greater than its length is necessary to prevent the vessel from getting too close to obstacles or even crossing them along the planned path. Additionally, the starting point and endpoint of the simulation should be within the range of the provided chart and located within the feasible area for the experimental vessel. Furthermore, the distance between the starting point and endpoint and the infeasible area should be greater than the length of the experimental vessel to ensure that the starting point and endpoint are not positioned within the grid of the obstacle, thus ensuring the feasibility of effective path planning.

1) The chart layering results are shown in Figure 7. The effective layers are as follows: the shoal area, the areas with water depths of 0–5, 5–10, 10–20, and more than 20 m.

2) Figure 8(a) shows the obstacle point identification results of vessel 1. The green part in the figure is an unfeasible area, including land, shoals, and regions with a water depth of 0–5 m. The water depth in other parts of the drawing is larger than 5 m, which can ensure the safe operation of vessel 1.

Figure 8(b) presents the obstacle point identification re-



(a) Shoal area

(b) Area with a water depth of 0–5 m

(c) Area with a water depth of 5–10 m

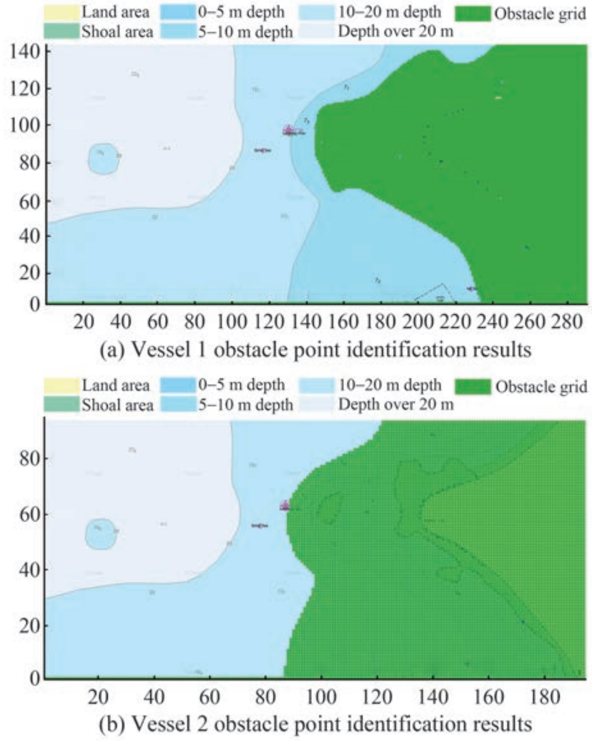
(d) Area with a water depth of 10–20 m

(e) Areas with a water depth of more than 20 m

Figure 7 Image layering results

sults of vessel 2. The green part in the figure is an unfeasible area, including land, shoals, and regions with water depths of 0–5 and 5–10 m. The water depth in other parts of the drawing is larger than 10 m, which can ensure the safe operation of vessel 2.

3) The chart gridding results of experimental ship 1 are shown in Figure 9(a). The actual length of the chart is 10 269.45 m, the actual width is 4 710.45 m, the number of horizontal grids is 289, and the number of vertical grids is 139. The length of each grid in the chart is 35.576 m, and the width is 34.092 m.

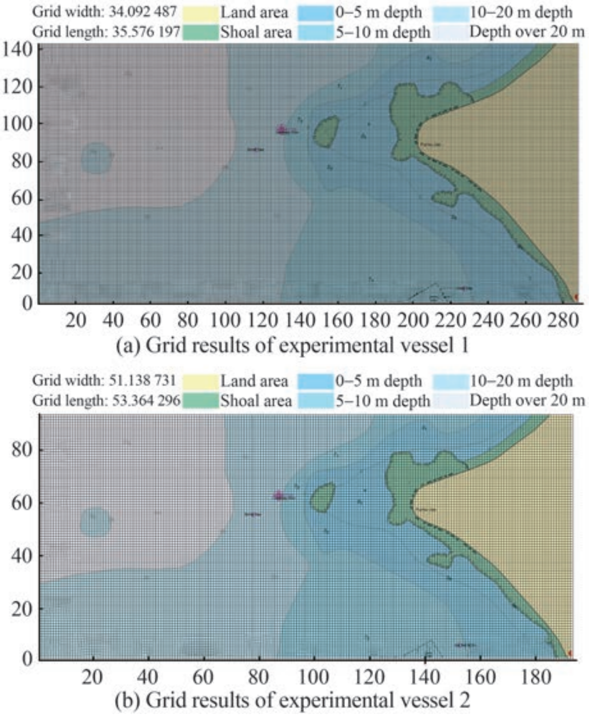


(a) Vessel 1 obstacle point identification results

(b) Vessel 2 obstacle point identification results

Figure 8 Obstacle Point Identification Results

The chart gridding results of experimental ship 2 are shown in Figure 9(b). The actual length of the chart is 10 269.45 m, the actual width is 4 710.45 m, the number of horizontal grids is 193, and the number of vertical grids is 93. The length and width of each grid in the chart are 53.364 and 51.139 m, respectively.



(a) Grid results of experimental vessel 1

(b) Grid results of experimental vessel 2

Figure 9 Grid results of experimental vessels

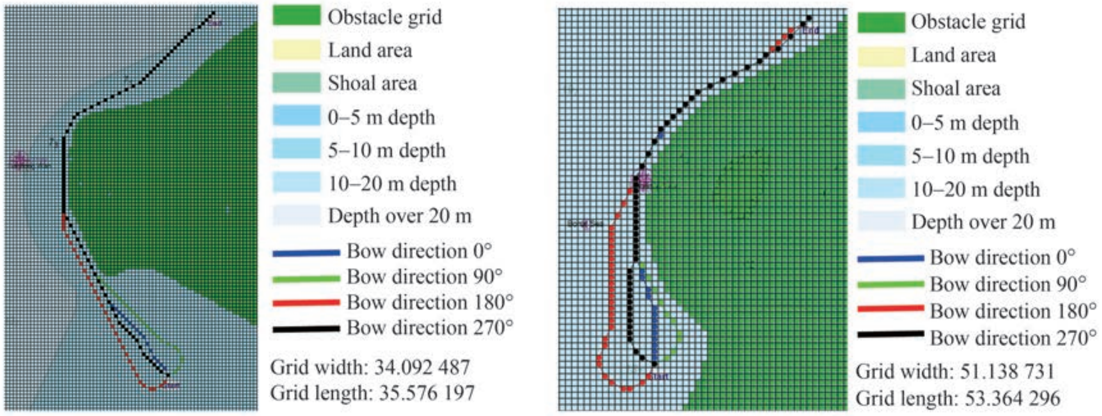
The grid results reveal that each grid size is close to the length of the ship, ensuring that the planned path does not pass through narrow areas that do not meet the requirements for the ship length while reducing the waste of obstacle-free areas in some obstacle grids.

4) Path planning results

① The path planning results for experimental ships 1 and 2 using the sparse A* algorithm for different initial ship directions are shown in Figure 10(a) and 10(b), respectively.

The blue, green, red, and black lines in the figure indicate the path planning results with an initial ship heading of 0°, the road planning results with an initial ship heading of 90°, the road planning results with an initial ship heading of 180°, and the road planning results with an initial ship heading of 270°, respectively.

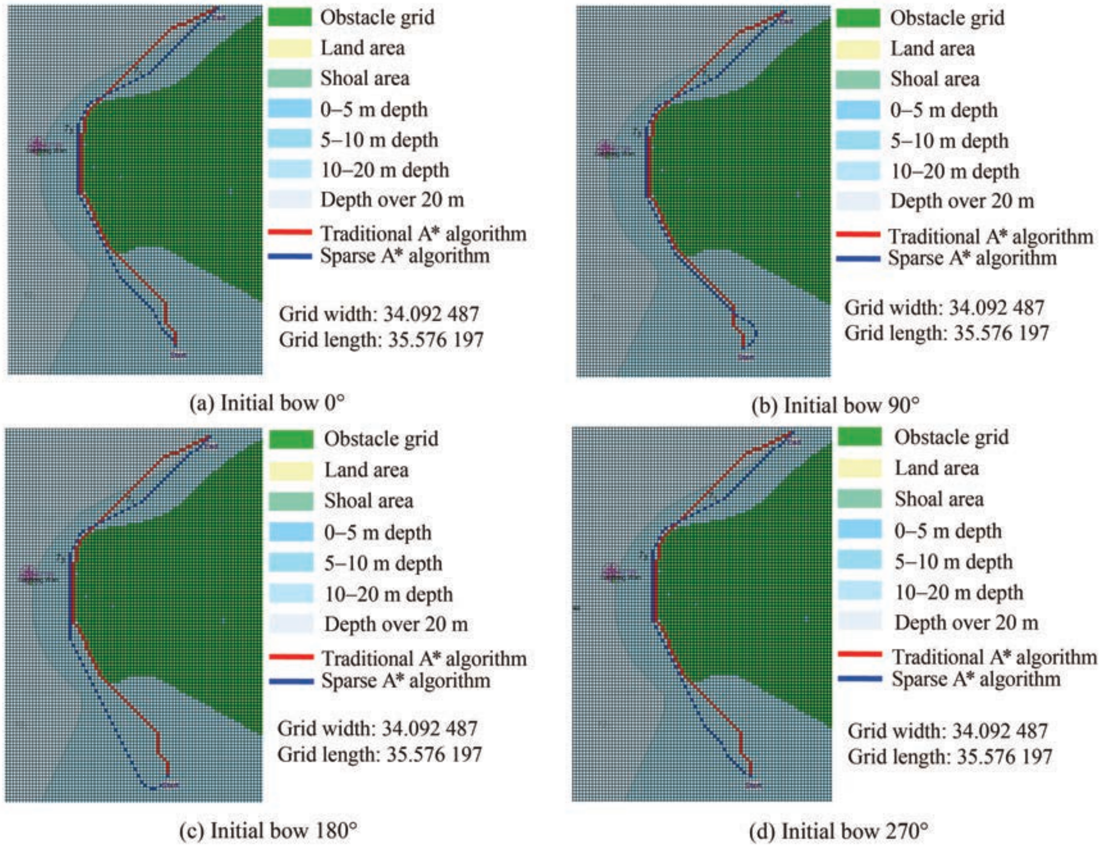
② The comparison of path planning results between the sparse A* algorithm for experimental ship 1 and the traditional A* algorithm is shown in Figure 11, and the comparison of path planning results for experimental ship 2 is shown in Figure 12.



(a) Results under different bow directions of vessel 1

(b) Results under different bow directions of vessel 2

Figure 10 Results under different ship heading conditions



(a) Initial bow 0°

(b) Initial bow 90°

(c) Initial bow 180°

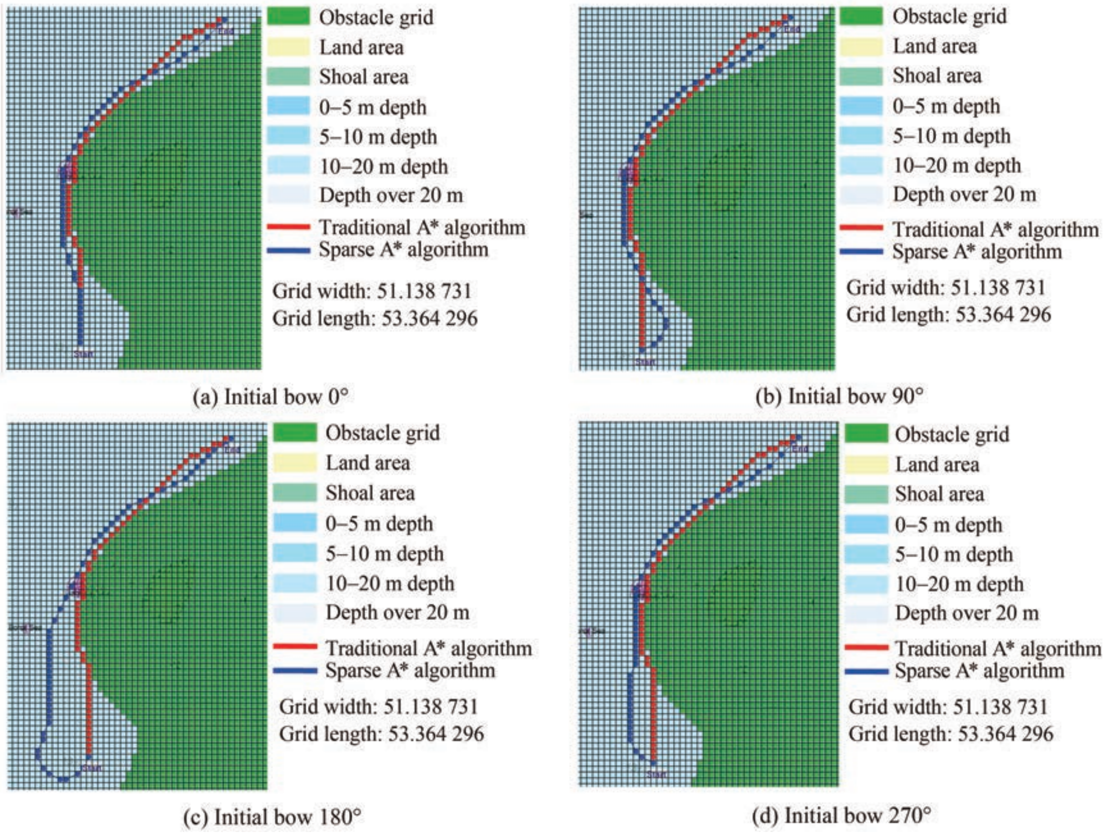
(d) Initial bow 270°

Figure 11 Comparison of path planning results for experimental ship 1

The red and blue lines in the figure denote the path planning results of the traditional A* and sparse A* algorithms, respectively.

5) The results of sparse A* and A* algorithms are compared, and the comparison results are presented in Table 1.

The table reveals that the average number of path points per unit length for the traditional A* algorithm is 0.02, while that for the sparse A* algorithm is 0.015, which is



(a) Initial bow 0°

(b) Initial bow 90°

(c) Initial bow 180°

(d) Initial bow 270°

Figure 12 Comparison of path planning results for experimental ship 2

Table 1 Comparison of algorithm performance

Experimental parameters	Planning algorithm	Path length (m)	Maximum number of storage nodes	Number of path points	Path average turning angle (°)	Times of passing through infeasible areas	Minimum distance from infeasible area (m)	Running time (s)
Test vessel 1 bow 0°	A*	4 835.65	3 136	113	165.405	50	24.614 8	1.979
	Sparse A*	4 720.2	2 421	85	177.361	0	61.490 8	42.741
Test vessel 1 bow 90°	A*	4 835.65	3 136	113	165.405	50	24.614 8	1.748
	Sparse A*	4 907.31	2 688	91	176.528	0	61.490 8	47.463
Test vessel 1 bow 180°	A*	4 835.65	3 136	113	165.405	50	24.614 8	1.873
	Sparse A*	5 017.37	3 218	88	176.835	0	61.490 8	62.298
Test vessel 1 bow 270°	A*	4 835.68	3 136	113	165.405	50	24.614 8	1.546
	Sparse A*	4 728.19	2 452	84	177.329	0	61.490 8	48.648
Test vessel 2 bow 0°	A*	3 706.64	503	61	166.271	33	36.922 3	0.513
	Sparse A*	3 688.29	321	48	173.152	0	92.236 1	4.571
Test vessel 2 bow 90°	A*	3 706.64	503	61	166.271	33	36.922 3	0.511
	Sparse A*	3 837.86	265	44	173.521	0	92.236 1	3.521
Test vessel 2 bow 180°	A*	3 706.64	503	61	166.271	33	36.922 3	0.528
	Sparse A*	4 389.53	554	54	172.722	0	92.236 1	7.609
Test vessel 2 bow 270°	A*	3 706.64	503	61	166.271	33	36.922 3	0.512
	Sparse A*	3 759.21	345	49	174.994	0	92.236 1	6.452

approximately 25% less than the traditional A* method. Additionally, the sparse A* algorithm has an average path turning angle that is approximately 10° greater than the traditional A* algorithm, resulting in a smoother path that is more conducive to ship tracking. The average maximum number of nodes consumed per unit path length by the traditional A* and the sparse A* algorithms is 0.392 and 0.324, respectively. This finding reveals a reduction of approximately 17% compared to the traditional A* method, resulting in a decrease in storage space usage. Moreover, the minimum distance between the planned path and the nonnavigable area calculated by the sparse A* algorithm is always higher than the set value and does not pass through nonnavigable areas. Comparison of the path planning results of experimental ships 1 and 2 reveals that the results of the sea chart grid vary with different ship conditions, leading to various planned paths. Furthermore, different draft depths and initial headings significantly affect the path planning effectiveness of the sparse A* algorithm. Compared with the traditional A* algorithm, the sparse A* algorithm markedly improves the safety and feasibility of the path.

In addition, when using the sparse A* algorithm in this study, the maximum expansion angle gradually increases by 5°, reducing the number of expanded nodes and the space occupied by the algorithm. However, these conditions also increase the running time of the algorithm to some extent.

5 Conclusion

This paper comprehensively considers the influence of the initial heading of the vessel, safety range, draft depth, and vessel length on the planned path. The sparse A* algorithm is employed to generate an optimal collision-avoidance route for the vessel from the current to the target point, which possesses high safety and practicality. Different vessel parameters and states are selected for simulation experiments, and the following conclusions are drawn: the average number of path points per unit length is reduced by approximately 25%, the average maximum expanded node count is decreased by approximately 17%, the average turning angle of the path is decreased by approximately 10°, and the required storage space is minimized. This approach is highly advantageous for tracking the path of the vessel. The minimum distance to infeasible areas always remains above the set value, and different paths can be planned for vessels with various initial headings, effectively improving the safety and feasibility of the planned route and reducing the possibility of vessel grounding or stranding accidents.

However, the algorithm used in this paper is currently only applicable to static environments. Thus, dynamic environmental factors, such as considering the influence of other ships on the planned path, should be incorporated into future work to increase the suitability of the improved algorithm for real maritime conditions.

Funding Supported by the Tianjin University of Technology Graduate R esearch Innovation Project (YJ2281).

Competing interest The authors have no competing interests to declare that are relevant to the content of this article.

References

Bi T, Ye P, Xu Y, Zhang F (2019) Route planning of unmanned aerial vehicle based on sparse A* algorithm. Proceedings of 2019 International Conference on Informatics, Control and Robotics (ICICR 2019), Shanghai, China, 31-37. DOI: 10.26914/c.cnkihy.2019.077835

Chen S, Liu C, Huang Z (2012) Global path planning for AUV based on sparse A* search algorithm. Torpedo Technology 20(4): 271-275. DOI: 10.3969/j.issn.1673-1948.2012.04.007

Chen X, Bose N, Brito M (2022) Risk-based path planning for autonomous underwater vehicles in an oil spill environment. Ocean Engineering 266: 113077.1-113077.13. DOI: 10.1016/j.oceaneng.2022.113077

Chen Z (2020) Using Gauss projection to realize band change of AutoCAD topographic map. Bulletin of Surveying and Mapping 525(12): 138-143. DOI: 10.13474/j.cnki.11-2246.2020.0409

Cheng X, Wang F (2023) Agv path planning based on improved A* algorithm. Computer Systems & Applications|Comput Syst Appl 32(3): 180-185. DOI: 10.15888/j.cnki.csa.009020

Du Y, Huang J, Zhang H (2021) Multi-direction path planning method of surface unmanned vechicle. Command Control and Simulation 43(4): 7-12. DOI:10.3969/j.issn.1673-3819.2021. 04.002

Feng H, Hu Q, Zhao Z (2022) AUV swarm path planning based on elite family genetic algorithm. Systems Engineering and Electronics 44(7): 2251-2262. DOI: 10.12305/j.issn.1001-506X. 2022.07.21

Grifoll M, Boren C, Castells-Sanabra M (2022) A comprehensive ship weather routing system using CMEMS products and A* algorithm. Ocean Engineering 255: 111427.1-111427.15. DOI: 10.1016/j.oceaneng.2022.111427

Hu S, Wu M, Shi J (2023) Research on improved A* algorithm integrating vector cross-product and jump point search strategy. Mechanical Science and Technology for Aerospace Engineering: 1-10. DOI: 10.13433/j.cnki.1003-8728.20230017

Lee T, Kim H, Chung H (2015) Energy efficient path planning for a marine surface vehicle considering heading angle. Ocean Engineering 107: 118-131. DOI: 10.1016/j.oceaneng.2015. 07.030

Li M, Zhang Y, Li S (2018) The gradational route planning for aircraft stealth penetration based on genetic algorithm and sparse A* algorithm. 2017 Asia Conference on Mechanical and Aerospace Engineering, 151: 1-5. DOI: 10.1051/matecconf/201815104001

Li J, Liu Q (2017) Dynamic path planning of unmanned aerial vehicle based on sparse A* algorithm and cultura algorithm. Journal of Applied Sciences 35(1): 128-138. DOI: 10.3969/j.issn.0255-8297.2017.01.014

Li X, Li H, Liu G, Bian S (2022a) Optimization of complex function expansions for Gauss-Krüger projections. ISPRS International Journal of Geo-Information 11(11): 566. DOI: 10.3390/ijgi11110 566

Li X, Ma X, Wang X (2022b) A survey of path algorithms for mobile robots. Computer Measurement and Control 30(7): 9-19. DOI: 10.16526/j.cnki.11-4762/tp.2022.07.002

Li Y, Duan P, Guo S (2022c) Overview of ship global path planning algorithms. Ship Standardization Engineer 55(5): 26-30+55. DOI: 10.14141/j.31-1981.2022.05.004

Li Y, Ma T, Chen P (2017) Autonomous underwater vehicle optimal path planning method for seabed terrain matching navigation. Ocean Engineering 133: 107-115. DOI: 10.1016/j.ocean-eng.2017.01.026

Li Z, Shi R, Zhang Z (2022d) A new path planning method based on sparse A* algorithm with map segmentation. Transactions of the Institute of Measurement and Control 44(4): 916-925. DOI:10.1177/01423312211046410

Liu Y, Huang H, Fan Q (2023) Based on improved A*_DWA algorithm for mobile robot path planning. Computer Integrated Manufacturing Systems: 1-20. http://kns.cnki.net/kcms/detail/11.5946.TP.20221125.1957.004.html

Miao C, Chen G, Yan C, Wu Y (2021) Path planning optimization of indoor mobile robot based on adaptive ant colony algorithm. Computers & Industrial Engineering 156(8): 107230-107230. DOI: 10.1016/j.cie.2021.107230

Miao T, El Amam E, Slaets P (2022) An improved real-time collision-avoidance algorithm based on hybrid A* in a multi-object-encountering scenario for autonomous surface vessels. Ocean Engineering 255: 111406.1-111406.15. DOI: 10.1016/j.oceaneng.2022.111406

Phanthong T, Maki T, Ura T, Sakamaki T, Aiyarak P (2014) Application of A* algorithm for real-time path re-planning of an unmanned surface vehicle avoiding underwater obstacles. Journal of Marine Science and Application 13(1): 105-116. DOI: 10.1007/s11804-014-1224-3

Qi J, Yang H, Sun H (2020) MOD-RRT*: A sampling-based algorithm for robot path planning in dynamic environment. IEEE Transactions on Industrial Electronics 68(8): 7244-7251. DOI: 10.1109/TIE.2020.2998740

Sang H, You Y, Sun X (2021) The hybrid path planning algorithm based on improved A* and artificial potential field for unmanned surface vehicle formations. Ocean Engineering 223: 108709.1-108709.16. DOI: 10.1016/j.oceaneng.2021.108709

Sedeno-noda A, Colebrook M (2019) A biobjective Dijkstra algorithm. European Journal of Operational Research 276(1): 106-118. DOI: 10.1016/j.ejor.2019.01.007

Singh Y, Sharma S, Sutton R (2018) A constrained A* approach towards optimal path planning for an unmanned surface vehicle in a maritime environment containing dynamic obstacles and ocean currents. Ocean Engineering 168:187-201. DOI:10.1016/j.oceaneng

Thi T, Copot C, Duc T, De K (2017) A hierarchical global path planning approach for mobile robots based on multiobjective particle swarm optimization. Applied Soft Computing 59(68-76): 68-76. DOI: 10.1016/j.asoc.2017.05.012

Wang S, Long T, Wang Z (2018) Dynamic path planning using anytime repairing sparse A* algorithm. Systems Engineering and Electronics 40(12): 2714-2721. DOI: 10.3969/j. issn. 1001-506X. 2018.12.14

Wei Y, Jin F, Dong K (2023) Improved global path planning A* algorithm based on node optimization. Computer Measurement and Control 31(6): 1-11. DOI: 10.3969/j. issn.1005-1228.2022. 02.002

Xie L, Xue S, Huang L (2018) Draught monitoring based on contour cluster analysis. Navigation of China 41(1): 60-63+108. DOI: 10.3969/j.issn.1000-4653.2018.01.012

Ying Z, He Q (2022) Unmanned vehicle path planning in complex waters based on improved A* algorithm. Mechanical & Electrical Technology 144(5): 33-35. DOI: 10.19508/j.cnki.1672-4801. 2022.05.010

Zhou S, Wang W, Tang J (2021) Improved sparse A* trajectory planning exploration incorporated with pheromone. Electronics Optics & Control 28(11): 26-30. DOI: 10.3969/j.issn.1671-637X. 2021.11.006

Zhou X, Zhou Y, Xu L (2020) A path planning algorithm for lunar cover based on probabilistic roadmap. Aerospace Control and Application 46(6): 43-49+78. DOI: 10.3969/j.issn.1674-1579. 2020.06.006

Zhu K (2022) Research and application of image segmentation based on improved FCM algorithm. Journal of Chongqing Technology and Business University (Natural Science Edition) 39(5): 24-33. DOI: 10.16055/j.issn.1672-058X.2022.0005.004