IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 15 January 2023, accepted 6 February 2023, date of publication 9 February 2023, date of current version 16 February 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3243810

APPLIED RESEARCH

A Method for Ship Route Planning Fusing the Ant Colony Algorithm and the A* Search Algorithm

YANFEI ZHANG, YIYAN WEN, AND HAIYANG TU
State Key Laboratory of Navigation and Safety Technology, Shanghai Ship and Shipping Research Institute, Shanghai 200135, China
Corresponding author: Yanfei Zhang (18862921828@163.com)

ABSTRACT Route planning has always been an essential issue in navigation research and an important manifestation of ship intelligence. In order to get the shortest route that meets the actual navigation requirements, this paper proposes a shortest path planning method based on Automatic Identification System (AIS) data, which establishes a high-precision environment model and combines ant colony algorithm (ACA) and A* search algorithm. We extract the key points from the initial route obtained by the A* search algorithm and then introduce the Bézier curve method to smooth the route to obtain the planned route. This strategy assures that the planned route satisfies the global optimal and actual navigation needs. A bulk carrier is selected for experimental validation, and the experimental results verify the effectiveness of the method proposed in this paper. Compared with the other algorithm, the algorithm proposed in this paper can obtain shorter paths faster and more efficiently when performed.

INDEX TERMS AIS data, ant colony algorithm, A* search algorithm, route planning, Bézier curve.

I. INTRODUCTION

Maritime transport is responsible for 90 percent of worldwide cargo transportation and plays a vital role in global economic development. Before setting sail, navigators must prepare a sensible route based on the ship's cargo capacity, speed, departure time, topographical data, and weather data, as the route directly affects the ship's safety and economy [1]. With the continual advancement of ship intelligence levels and unmanned ship, automatic route planning has become a fundamental technology for autonomous ship navigation, with the level of route planning determining the intelligence level of the ship [1], [2], [3], [4]. Artificial intelligence technology, big data mining technology, and electronic information technology enable the gradual replacement of human route drawing with the utilization of existing route resources to produce more appropriate route proposals.

In recent years, numerous academics have presented a variety of route planning algorithms that considerably enhance the performance of route planning algorithms and make the automatically generated routes superior, more cost-effective, and dependable, with a more significant practical application value [5], [6].

Some academics rasterize electronic navigational charts (ENC) and construct grid environment models using obstacle extraction or obstacle grid setup. Wang et al. [7] developed the binary trees method based on the ENC, while Lyu et al. [8] proposed the stack method to bypass the barriers on the way back. Wang [9] incorporated the route's width into the algorithm so that the ship could avoid obstructions more practically and achieve automatic route planning. Guo [10] used the grid method to process the marine environment in the ENC into a two-dimensional simulation of the marine environment and used a deep Q network (DQN) based optimization algorithm to plan the coastal route automatically.

In addition, Automatic Identification System (AIS) data-based automatic route planning is a popular study area. Many researchers have used route planning algorithms such as the Dijkstra algorithm [11], [12], dynamic planning algorithm [13], [14], ant colony algorithm (ACA) [15], and its optimization algorithm to generate optimal routes based on the navigable points or trajectory points in the AIS data [16].

Numerous researchers have also incorporated other algorithms to improve the route planning approach. Andersson and Ivehammar [17] devised a dynamic route planning

The associate editor coordinating the review of this manuscript and approving it for publication was Yiqi Liu.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

algorithm to compute the maximum benefit of a route; their method can assist ship owners in reducing fuel consumption expenses and CO_2 emissions. Their proposed strategy can assist ship owners in reducing fuel expenses and carbon dioxide emissions. Dong et al. [18] modified the conventional ant colony algorithm and suggested a double ant colony algorithm to create an energy-efficient route planning system. Li et al. clustered the trajectory lines of AIS, with their suggested clustering technique employing the merge distance theory, and the improved DBSCAN algorithm able to determine the trajectory's direction [19]. The strategy suggested by Peng begins by performing noise-based density-based spatial clustering on a large number of vessel trajectories to construct distinct clusters of trajectory vectors. Then, calculate their centerlines iteratively in the clusters of trajectory vectors and design a canal network based on the node-arc topology link between these centerlines [20]. Zhang [21] accurately determined many turning sites by evaluating a significant amount of AIS trajectory data and then planned routes using an ant colony method. Both Liu [22] and Song [23] optimized the traditional A* algorithm in the process of route planning for unmanned vessels and gridded the map to obtain routes in complex obstacle environments finally.

However, relevant research and technology still need to resolve issues. First, the automatic ocean route planning algorithm based on electronic charts must rely on many electronic chart data [7], [8], [9]. Wang and Lyu's [7], [8] paper also mentions that further improvements in accuracy require access to more chart information, such as obstacles. Therefore, the latest ENC data needs to be updated in stages, and dynamically updating these electronic chart data incurs higher economic and time costs than AIS data. Secondly, whether based on ENC or AIS data, automatically planning routes must consider the environmental model's accuracy [7], [16], and the increase in accuracy will increase computational cost. However, if the precision of the environment model is lowered, the automatically generated path will cross land, which is dangerous and unusable. Thirdly, ACA and the Dijkstra algorithm can locate the globally best path, but the calculation time is lengthy [15].

Therefore, in response to the gap proposed in the previous paragraph, this paper proposes an automatic shortest-route planning method based on AIS data. The proposed method combines ACA with A* search algorithm to determine the shortest path. First, ACA restricts the initial shortest route region range, which can also reduce large calculation nodes. Then, we take advantage of the fast search speed of the A* algorithm to look for the shortest path within the initial range constraints. Finally, we optimize the projected route line by removing duplicate points, extracting key points, and utilizing the Bézier curve method to smooth the route.

The remaining sections of this paper are as follows: Section II introduces the process of constructing the environment model; Section III proposes the shortest path planning method combining ACA and the A* search algorithm; in Section IV, a bulk carrier is utilized as the target vessel to plan a part of its route with the method provided in this paper and compare it to the actual trajectory of the vessel to illustrate the algorithm's applicability. Section V provides a summary of the entire paper.

II. ENVIRONMENT MODEL DEVELOPMENT

A. PRE-PROCESSING AIS DATA

Before developing the environment model, one of the most crucial processes is pre-processing the vast quantity of AIS data. The AIS data selected for this study are mainly from the sea off China, totaling about 35,000 items. First, we cleaned the abnormal AIS data. The principles of excluding abnormal data are as follows:

- The longitude $< -180°$ or $> 180°$;
- The latitude $< -90°$ or $> 90°$;
- The sailing speed < 0 or > 50 knots;
- The depth < 0 meters;
- The heading $< 0°$ or $> 360°$;

After the first step of processing, about 34,000 items of data can be used for the subsequent study.

Second, the parameters, including latitude, longitude, speed, and draft data from the AIS data, were kept for the developing environment model. Filtering ship speed ensures that the environmental model is constructed to meet the speed requirements of the ship and that no obstacle areas are included in the environmental model. The filtering of the ship's draught ensures that the environmental model is constructed to meet the draught requirements of the ship and that the route is planned within the allowable water depth for the target ship. Finally, it is necessary to filter the AIS data ranges based on the target vessel's speed and actual draft further.

B. ENVIRONMENT MODEL DEVELOPMENT

This section employs the grid mapping approach to construct the environment model using the AIS data filtered in Section II-A. A ship's route in a particular navigation area may be simplified to the path of a ship between two locations on a plane. The grid mapping method breaks the supplied navigation area map into some grids and assigns the specified unit length to each grid's edge length. The free attribute grids designated with white color and the obstacle grids marked with black color are then decided based on the navigability of a single grid in the navigation area map. Using the grid mapping method, the route planning issue in a complex spatial navigation environment is converted into searching the path between two grid centroids in a two-dimensional grid that satisfies certain constraints.

Assuming that the length of the working environment map is L, the breadth is N, each square grid has a height and width of l, the environment may be partitioned into an information map of nn^*mm cell grids, satisfying the Equation (1).

$$l = L/nn = B/mm \tag{1}$$

In route planning, AIS point data in a grid is used to determine a free attribute grid; if there is no AIS data, the grid is an obstacle grid.

Chart represents the whole environment, $grid_{i,j}$ represents all the information of a grid unit, i denotes horizontal, *i*.e., east-west direction, and j denotes vertical, i.e., north-south direction, all grids are placed in the set P, n and m are the maximum values of i and j in the environment space, and the expression is shown in Equation (2)

$$\begin{aligned} P &= \{(i,j) | 1 \leq i \leq n, 1 \leq j \leq m\} \\ Chart &= \{grid_{i,j} | grid_{i,j} = 0 \text{ or } 1, (i,j) \in P\} \end{aligned} \tag{2}$$

III. METHODOLOGY

The proposed method consists of three steps. In the first step, the estimated grid range of the shortest path is determined. Utilize ACA to calculate the range through which the shortest path goes on the low precision environment model. Second, locating the shortest route in an improved environment model. This step will rebuild a high-precision environment model based on the first step's approximate grid range and calculate the shortest route using the A* search duplicate points, extract critical points, and then smooth the path using the Bézier curve method. The flow chart is depicted in Figure 1. Thirdly, optimize routes, so the planned path corresponds to the actual route.

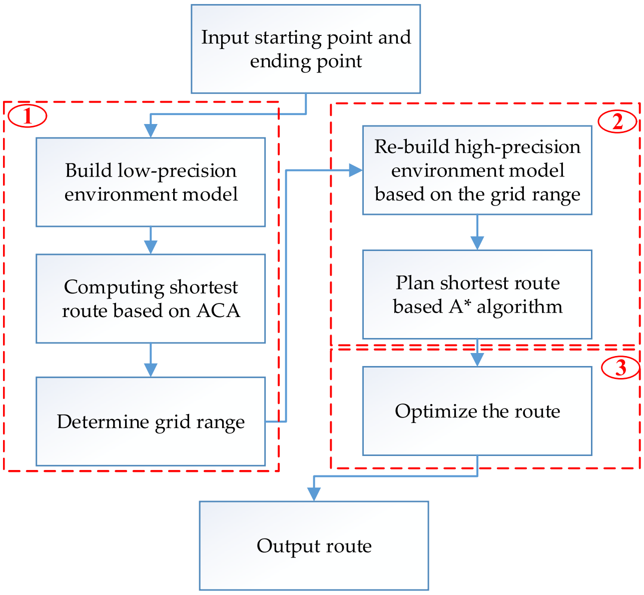
Input starting point and ending point
1
Build low-precision environment model
Computing shortest route based on ACA
Determine grid range
2
Re-build high-precision environment model based on the grid range
Plan shortest route based A* algorithm
3
Optimize the route
Output route

FIGURE 1. The flow chart of route planning model.

A. DETERMINE THE GRID RANGE WITH THE LEAST DISTANCE USING ACA

This section's primary purpose is to determine the grid range of the shortest route. First, the low-precision environment model is generated, the grid width is set to 0.5°, and ACA is used to determine the shortest route under the low-precision environment model. The free attribute grids in eight directions of each node along the route are then included in the grid range, while the free attribute grids and obstacle grids in other places are classified as obstacle grids.

1) CONSTRUCTING THE ADJACENCY MATRIX

An environmental model-compliant adjacency matrix is utilized to record the distance between adjacent navigable grids, as shown in Figure 2. As depicted in Figure 2, $grid_{i-1,j+1}$ and $grid_{i,j-1}$ are obstacle, thus, ant in $grid_{i,j}$ can travel to other six free attribute grids, the distance between two free attribute grid is given by Equation (3), and the distance between free attribute grid and obstacle grid is infinity.

$$\begin{aligned} d = \arccos(&\sin(lat_{i,j}) * \sin(lat_{i,j+1}) + \cos(lat_{i,j}) \\ &* \cos(lat_{i,j+1}) * \cos(lon_{i,j+1} - lon_{i,j})) \end{aligned} \tag{3}$$

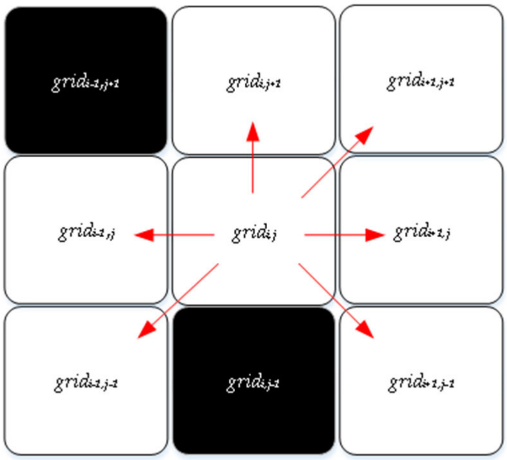

FIGURE 2. Neighborhood grid schematic.

where $lat_{i,j}$ denotes the latitude of $grid_{i,j}$; $lon_{i,j}$ denotes the longitude of $grid_{i,j}$; $lat_{i,j+1}$ denotes the latitude of $grid_{i,j+1}$; $lon_{i,j+1}$ denotes the longitude of $grid_{i,j+1}$.

2) ANT COLONY ALGORITHM (ACA)

ACA is an intelligent evolutionary algorithm based on parameterized probability distribution, where the model parameters are the pheromones generated by the individual ant colony; hence, ACA is, in essence, an intelligent pheromone-based optimization method. The core of ACA for ship route planning is the update of state transfer probability and pheromone [15]. Assume that the total number of ants is m. The transfer probability (p_{ij}^k) from route node i to route node j at time t for the k_{th} ant is given by Equation (4).

$$p_{ij}^k = \begin{cases} \dfrac{\tau_{ij}^{\alpha}(t)\eta_{ij}^{\beta}(t)}{\sum_{j \in allowed_k} \tau_{ij}^{\alpha}(t)\eta_{ij}^{\beta}(t)}, & j \in allowed_k \\ 0, & j \notin allowed_k \end{cases} \tag{4}$$

$$\eta_{ij}(t) = 1/d_{ij} \tag{5}$$

where the parameter α is information heuristic factor, indicating the importance of pheromone concentration. The parameter β is expectation heuristic factor, indicating the importance of the distance between the route nodes. The symbol $\tau_{ij}(t)$ is the pheromone concentration between the route nodes i and j at time t. The symbol $\eta_{ij}(t)$ is the heuristic function, which represents the expectation of an ant individual from the route node i to j, takes the value $1/d_{ij}$, as Equation (5) shown, d_{ij} is the distance between the current node i and the next node j.

If the value of the distance d_{ij} between the current node and the next node is larger, the value of the heuristic function $\eta_{ij}(t)$ will be smaller; consequently, the probability of ants choosing this node will be smaller, as will the transfer probability; conversely, if the value of the distance d_{ij} between the current node and the next node is smaller, the value of the heuristic function $\eta_{ij}(t)$ will be larger. Therefore, the distance between the present node and the next node influences, to some extent, the probability that this ant will choose that node. In order to prevent the ants from selecting the same nodes multiple times, a taboo table is created and updated whenever the ants reach a new node. The symbol $allowed_k$ is the turn point of the path that the k_{th} ant can pick during the search process, and the transfer probability takes the value 0 if it is not within this range. This assures that the ants will not continually pass a node and form a loop. Consequently, the ants select numerous possible paths based on the pheromone and the length of the path.

According to the idea of ACA, the algorithm achieves optimal path solution by constantly passing ants via a shorter path over time, while leaving pheromones on this road to attract more ants. Due to the existence of this positive feedback mechanism, if the pheromone on this shorter path is too strong, the heuristic information of the ant colony will lose its effectiveness and the ants will not explore other solutions in the space, causing the ant colony algorithm to reach a local maximum. To remedy this issue, a volatile mechanism must be added to the ant colony algorithm to update the pheromone, and the pheromone is updated after each iteration based on this volatile mechanism. Therefore, the pheromone on the path at the moment $(t + 1)$ is updated in the manner shown in Equation (6) and (7).

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \Delta\tau_{ij}(t) \tag{6}$$

$$\Delta\tau_{ij}(t) = \sum_{k=1}^{m} \Delta\tau_{ij}^{k}(t) \tag{7}$$

We chose Ant-Cycle model to update the pheromone. Equation (8) represents the update of pheromone using global information, i.e., when the ant reaches the end point, the algorithm updates the pheromone on the whole path searched out this time.

$$\begin{aligned} \Delta\tau_{ij}^{k}(t) &= \frac{Q}{L_k}, \quad \text{if the } k^{th} \text{ ant reaches node } j \text{ from} \\ &\qquad \text{node } i \text{ in this iteration.} \\ \Delta\tau_{ij}^{k}(t) &= 0, \quad \text{if the } k^{th} \text{ ant does not reach node } j \text{ from} \\ &\qquad \text{node } i \text{ in this iteration.} \end{aligned} \tag{8}$$

where the symbol $\tau_{ij}(t+1)$ is the pheromone of the route node i to node j at time $t + 1$. The symbol $\Delta\tau_{ij}^{k}(t)$ is the increment of pheromone released by the k^{th} ant on the path at time point t, and the initial value is 0. The parameter ρ is The pheromone volatility factor, taking values between [0, 1] and $1-\rho$ is the pheromone residual factor. The parameters Q is the pheromone intensity. The parameter Lk is the length of the route traversed by the k^{th} ant in this iteration.

As shown in the Equation (7) and (8), the heuristic function, the pheromone, and the collaboration among ants all have a significant impact on the convergence of the algorithm; in this case, the pheromone correlation factor α, $(1-\rho)$, β, the number of ants m, the pheromone strength Q, and the iteration number in the model are also crucial factors that affect the algorithm's performance and efficiency [25].

3) DETERMINE THE GRID RANGE WITH THE LEAST DISTANCE

We apply ACA algorithm to determine the shortest path given the present environment model, with the primary procedure being as follows.

1) Convert the specified starting point A and ending point B to their respective grid positions in the grid map, starting grid A and ending grid B;
2) Initialize and set the parameters for each ant population;
3) Select the next neighboring free attribute grid using the roulette wheel approach. The adjacent free attribute grid and its distance are determined based on the findings of the free attribute grids partitioning model;
4) Each ant follows the pheromone's trail until it reaches ending point B;
5) Document the passable path and total distance traveled by each grid;
6) Update the pheromone;
7) Determine if the population has reached its maximum iteration;
8) Return the shortest route;

Figure 3 is the flow chart of finding the shortest path based on ACA.

Second, the free attribute grid range is further reduced according to the calculated shortest path. The five grid nodes through which the shortest route travels, as depicted in Figure 4.a, the free attribute grid adjacent to each node is within the range determined by the shortest path. As shown in Figure 4.b, the blue grid is the range of grids chosen by the shortest route, while grids outside of this range become obstacle grids.

Finally, we employ ACA to determine the shortest path's grid range under low-precision environment model.

B. SHORTEST ROUTE PLANNING MODEL BASED ON A* SEARCH ALGORITHM

This section employs the A* search algorithm to compute the shortest route in the exact environment based on the grid range involved in the shortest path found in section III-A

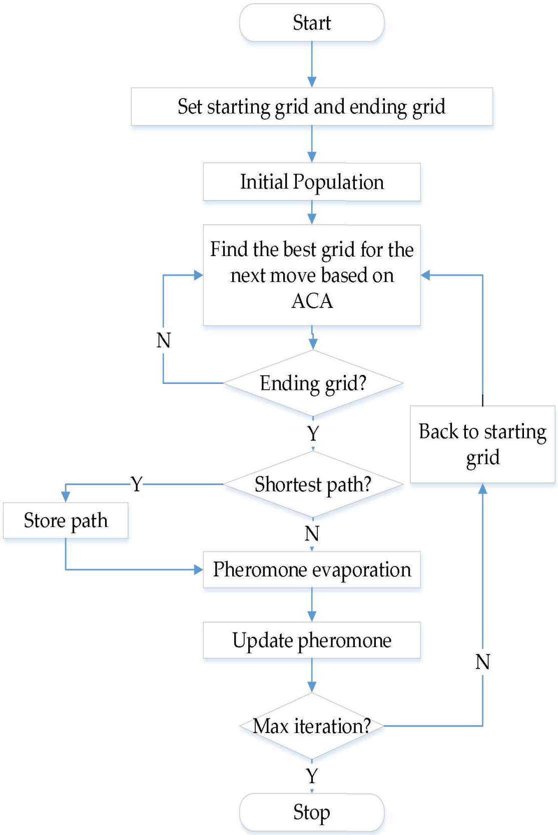

FIGURE 3. The flow chart of finding the shortest path based on ACA.

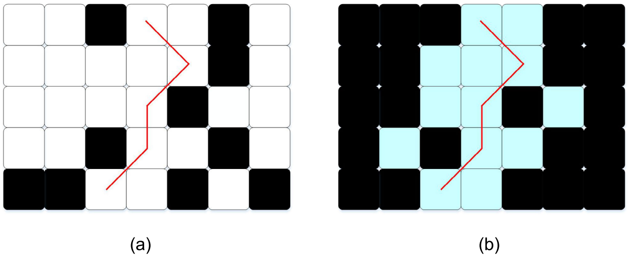

FIGURE 4. Schematic diagram of the grid range of the shortest route: (a) Original route; (b) Grid range map.

to assure the route's safety. In the first step, the high-precision environment model is produced under the grid range initially defined in Section III A. In the second step, the A* search algorithm is applied to find the shortest path in the high-precision environment model.

The grid range provided by ACA is a somewhat coarse model, and we must still improve the grid to provide a more precise environment model. The width of the unit grid is lowered to 0.0625° to generate a more accurate environment model based on the navigable range determined in Section III-A. The grid width of this environment model is 1/8 of the grid width of the section III-A environment model.

The A* search algorithm chooses the route that minimizes Equation (9), a heuristic function of the A* search method.

$$F(n) = G(n) + H(n) \tag{9}$$

where $G(n)$ is the true cost of moving between the starting nodes of the current grid, as demonstrated by Equation (10), while $d(n, n-1)$ is given by Equation (3). And $H(n)$ represents the predicted movement cost between the current and end grid. The A* search algorithm and Dijkstra algorithm are identical if $H(n) = 0$ [26]. The current grid's distance function to the ending grid is substituted for $H(n)$ in Equation (3).

$$G(n) = d(n, n-1) + G(n-1) \quad (n > 1) \tag{10}$$

A* search algorithm uses the open list to hold nodes waiting to be examined and the close list to store nodes that have been previously explored. The A* search algorithm follows these steps:

Step 1: Insert the starting point S into the open list;

Step 2: If the endpoint E is present in the open list, the path search is complete. If no open list exists, the path does not exist;

Step 3: Choose the node u in the open list with the most minor estimated value F as the active node and add it to the close list;

Step 4: Obtain all of the directly accessible nodes, and then update the open list. if v is in the close list, do not process; if v is not in the open list, add v to the open list, and its associated $F = G(u) + \text{distance}(u, v) + H(v)$; if v is in the open list, check for a smaller F value for v and update the F value of v if there is a smaller F value.

Repeat Step 2 to 4 until the process is complete.

Finally, the each node of shortest route is output.

C. ROUTE OPTIMIZATION BASED ON BÉZIER CURVE METHOD

The A* search algorithm's suggested path includes numerous redundant steering points and sharp turning points significantly different from the actual route. To ensure that the designed route meets the exact turning radius of the ship, we introduce the bezier method of smooth route planning so that the vessel does not change direction sharply at a specific turning point and affect the standard navigation of the ship. For this issue, we presented a Bézier curve-based path optimization approach. The main steps are as follows.

Firstly, eliminate redundant points and extract key points. Suppose the current path node is on the same line as the preceding and subsequent nodes. In that case, the current node is deleted, the path is updated, the subsequent nodes are found to be on the same line, and the last remaining node is the critical point. Furthermore, extract key points and ensure that the reconnected line segment does not cross the obstacle grid. For instance, after the first step, the key points $(M_1, M_2, \ldots, M_k)$ are gathered, and M_1 and M_3 are linked to generate the line l (The line function: $Ax+By+C = 0$). Determine whether the connected M_1 and M_3 will pass through the obstacle grid N ((x, y), $(x, -y)$, $(-x, -y)$, $(-x, y)$ for the four vertices of the obstacle grid N) using the Equation (11). If it does not pass through the obstacle grid N, point P_2 can be deleted, the path can be updated, and subsequent M_k and M_1 can be joined in succession until the new line segment goes through the obstacle grid N. The final processed route $\{P_i | i = 1, 2, 3, \ldots, \mathrm{n}\}$ is obtained.

$$\begin{aligned} &(A(-x) + By + C)(Ax + B(-y) + C) > 0 \\ &(A(-x) + B(-y) + C)(Ax + By + C) > 0 \end{aligned} \tag{11}$$

Secondly, the processed route is smoothed using the third-order Bézier curve method, as shown in Equation (12). Given the coordinates of control points P_i to P_{i+3}, the first point $P_i = \{x_i, y_i\}$, the second point $P_{i+1} = \{x_{i+1}, y_{i+1}\}$, and so on. The parameter t moves from 0 to 1, the step is 0.05, and the loop goes over 0, 0.05, 0.1, 0.15, ..., 0.95, 1.

$$\begin{aligned} P = P_i * (1-t)^3 + 3P_{i+1} * t(1-t)^2 \\ + 3P_{i+2} * t^2(1-t) + P_{i+3} * t^3 \end{aligned} \tag{12}$$

Figure 5 briefly illustrates the route after the two-step process. Figure 5.a is the initial route generated by the A* search algorithm, Figure 5.b shows the route after eliminating redundant points, Figure 5.c shows the route after extracting key points, and Figure 5.d shows the route after smoothing with the Bézier curve method.

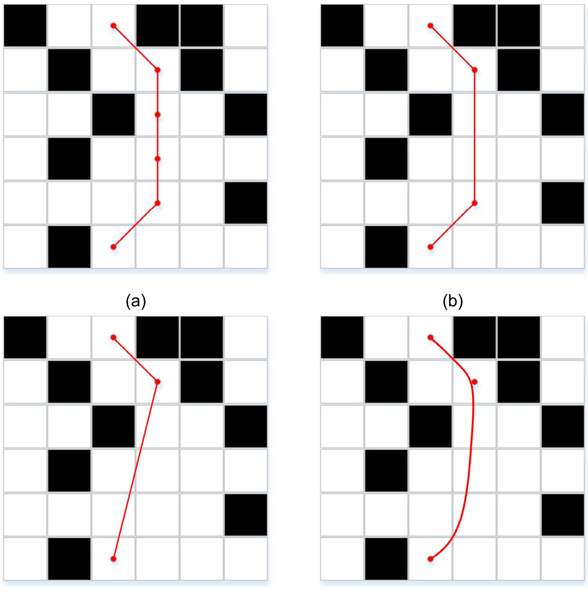
(a)
(b)
(c)
(d)

FIGURE 5. Route optimization process diagram: (a) Initial route after A* search algorithm calculation; (b) The route after eliminating redundant points; (c) The route after extracting key points; (d) The final route after Bézier curve smoothing.

IV. EXPERIMENTAL RESULTS AND DISCUSSION

A. EXPERIMENTAL DESIGN

This experiment's target vessel is a bulk ship whose major parameters are detailed in TABLE 1. The vessel sails from the west coast of Africa via the Cape of Good Hope and the Strait of Malacca to the Chinese city of Tianjin. A portion of the ship's itinerary was chosen as the subject of this investigation. The ship set sail for the sea of Zhoushan Islands, China, on June 4, 2022, at 13:00 (UTC), when the longitude was 123.981°E, the latitude was 30.875°N, and the port berth was located at 118.485°E and 38.826°N. Accordingly, we confined the search area to latitudes 30°N to 41°N and longitudes 117°E to 125 E. In addition, we kept AIS data with a depth of 10m or more and a speed of more than 5kn based on data from TABLE 1 and principle in Section II-A.

The code used in this study both written in python, and the program was run on a particular laptop with 4 Core i5 for testing.

TABLE 1. The experimental ship's detailed parameters.

Feature	Value
Ship Type	Bulk ship
Displacement	210,947(t)
Propeller Diameter	9600(mm)
LBL	299.9(m)
B(mld)	50(m)
Design draft	16.1(m)
Design speed	14.5(kn)

B. EXPERIMENTS RESULTS AND ANALYSIS

Firstly, determining the shortest route's grid range. Moreover, the grid width is set to 0.5° in this step as mentioned in Section III-A. As a result, the low-precision environment model consists of 16*22 grids, as depicted in Figure 6, where white represents the free attribute grid and black represents the obstacle grids. The inital values of ACA are given in TABLE 2, which is referred from Wu et al. paper [1].

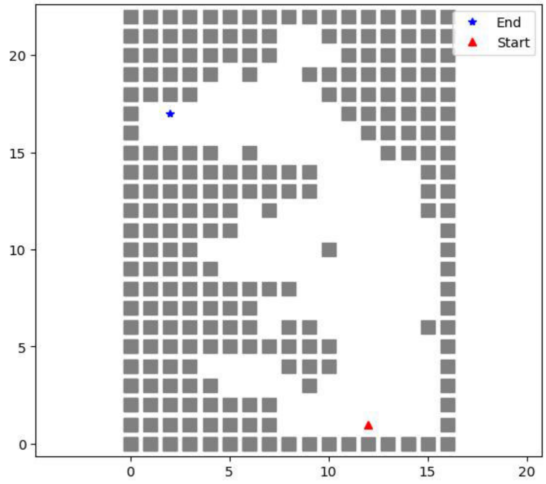

FIGURE 6. The low-precision environment model for the ACA.

Figure 7 depicts the shortest grid region calculated by the ACA. The red line in the illustration represents the route determined by the ACA. Compared with Figure 6, Figure 7 demonstrates that after the initial optimization search by

TABLE 2. The main parameters of the ACA.

Parameter name	Value
The number of ant	20
The iteration	500
The pheromone heuristic factor a	1
The expectation heuristic factor b	7
The pheromone volatility factor r	0.3
The pheromone intensity Q	1

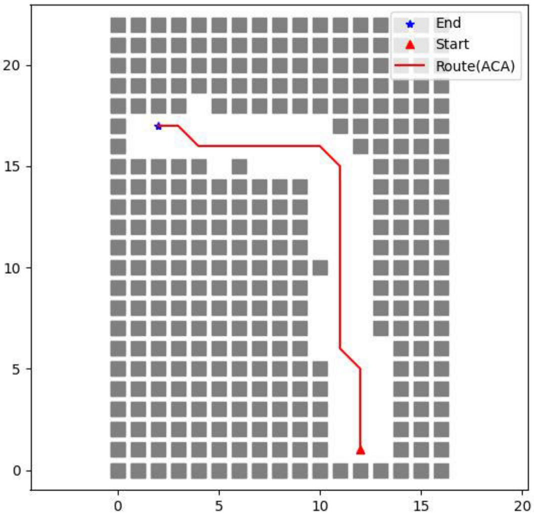

FIGURE 7. The route and grid range calculated by the ACA.

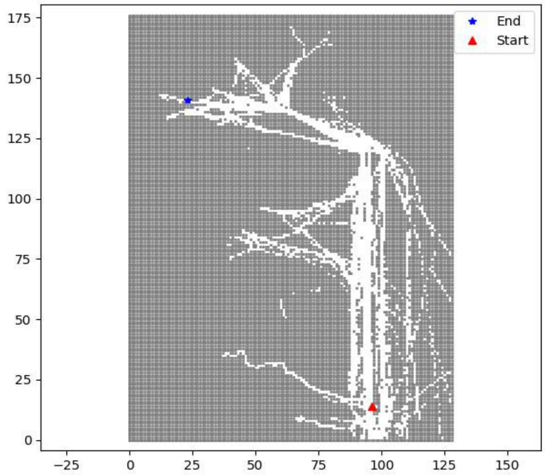

FIGURE 8. The high-precision environment model for A* search algorithm.

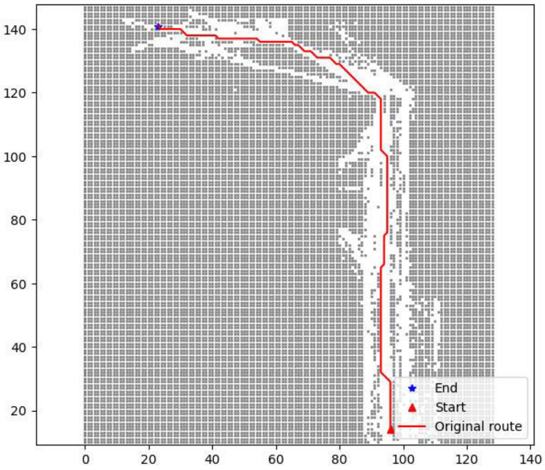

FIGURE 9. The route calculated by A* search algorithm.

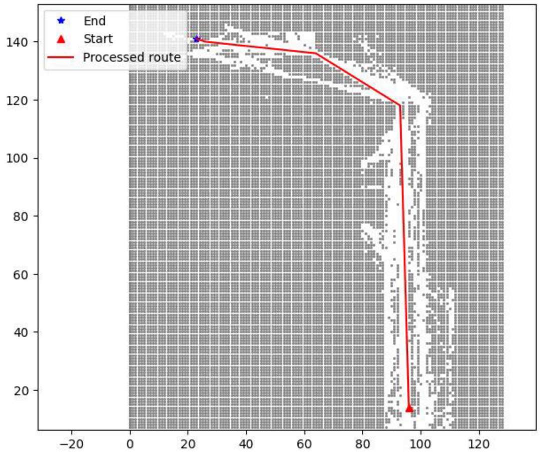

FIGURE 10. The route after removing unnecessary points and extracting key points.

the ACA, the navigable area is reduced further. This step restricts the initial shortest route region range and reduces the number of search nodes, improving search efficiency and saving search time for subsequent searches.

In this stage, the A* search algorithm was utilized to determine the shortest route based on the navigable grid range determined in the previous step. As illustrated in Figure 8, we began by refining the environment model by reducing the grid width to 0.0625° to obtain a more precise environment model. As a result, the high-precision environment model consists of 128*176 grids.

Based on the grid range obtained by the ACA, we reconstructed the grid depicted in Figure 8 and use the A* search algorithm to determine the shortest route to acquire the results depicted in Figure 9; the red line represents the A* search algorithm's computed route. After the initial determination of the shortest route range by the ACA algorithm, the number of free attribute grids decreased from 3172 to 2108. More importantly, a comparison of Figure 8 and Figure 9 demonstrate that the range of free attribute grids is significantly reduced, free attribute grids that are not related to the route were excluded.

According to the method described in Section III-C, the red path in Figure 9 was optimized, with a total of 176 turning points. Firstly, we obtain the processed route depicted in Figure 10 by eliminating the unnecessary points and extracting the key points, five turning points remaining. Then, we used the Bézier curve method to smooth the processed route depicted in Figure 10, resulting in the optimal route depicted in Figure 11, and the final 45 points were generated. We used Equation (11) to check the safety of the 45 points shown in Figure 11. None of the 44 line segments made up of the 45 points pass through the obstacle and can be sailed safely. The total distance of the three routes is calculated as indicated in TABLE 3. Following optimization, the total distance of the routes is decreased by 12.8 nautical miles. The difference between the distance of the route before and after smoothing is around 0.05 nm. Selecting some of the segments for evaluation, as shown in Figure 12, the original route's meandering route requires only two crucial locations to reach the ending point. Therefore, this can further prevent the rise in navigation miles produced by the A* algorithm's local optimum. As illustrated in Figure 13, we also employed the Bézier curve method to smooth the route's turning points, making the planned route close to the real navigation.

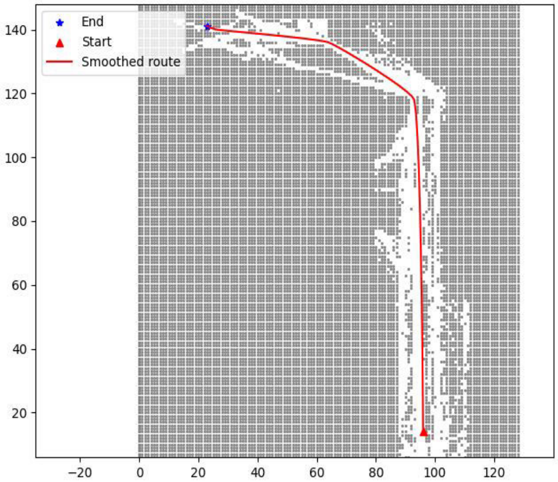

FIGURE 11. The route after Bézier curve smoothing.

TABLE 3. The total distance for each route.

Route	Total distance (nm)	Number of turning points
Initial route (ACA + A* search algorithm)	635.12	176
Processed route (Extract key points)	622.27	5
Smoothed route (Bézier curve method)	622.32	45

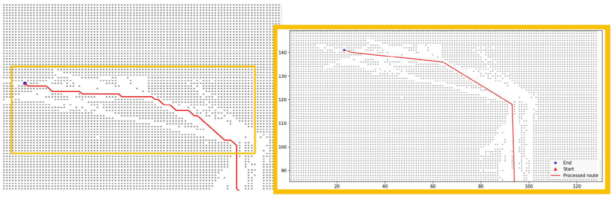

FIGURE 12. The comparison of the original route and the processed route.

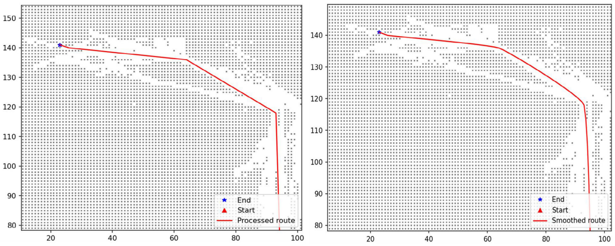

FIGURE 13. The comparison of the processed route and smoothed route.

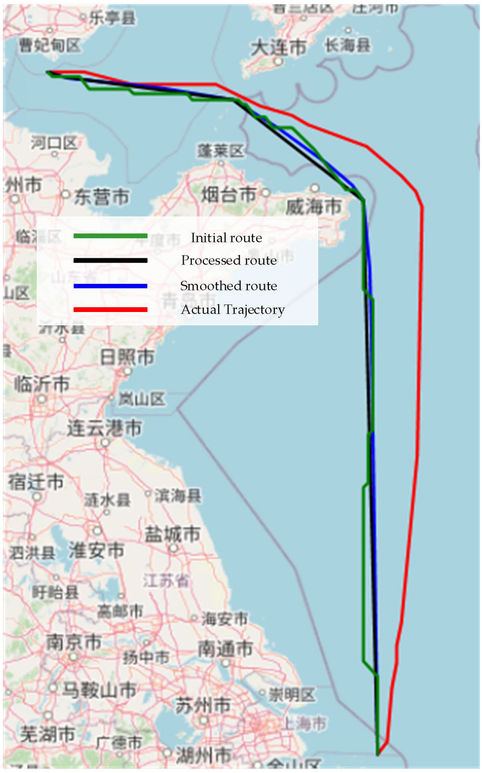

FIGURE 14. The comparison of the planned route and the actual trajectory.

To verify the effectiveness the proposed algorithm, we compared the planned route proposed by in this paper and actual trajectory of the target ship, as depicted in Figure 14. It can be observed that after passing through the Zhoushan Islands, the target ship did not take a straight line to the north but instead slightly changed its direction and sailed towards the outer sea before heading north to the Bohai Sea. As a result, the total distance of the actual trajectory is 662 nautical miles. Compared to the planned route, the distance of the actual trajectory is approximately 40 nautical miles longer than the planned route. In addition, the route safety check demonstrates that all planned routes meet navigational safety requirements and allow transit through waterways safely. The planned route in the figure also indicates that the planned path safely traverses the Bohai Strait via the Old Tieshan waterway before arriving at its destination, which is identical to the actual route.Therefore, the planned routes obtained with the algorithm proposed in this paper are able to meet the route safety requirements.

To verify the advancement of proposed algorithm, we compared it with A* algorithm. Figure 15 shows the result of A* search algorithm, and Table 4 shows the planned route length and planning time. From Table 4, The algorithms proposed in this paper all outperform the results calculated by the comparison algorithms. Compared with the single A* search algorithm results, the proposed algorithm's planning time is reduced by about 42%. The length of the planned route is reduced by about 2.5%, and the number of turning points is reduced by about 74.5%, so the proposed algorithm can significantly improve the search efficiency and plan a shorter route.

TABLE 4. Comparison of the proposed algorithm with the A* algorithm.

Algorithm	Route length (nm)	Planning time (S)	Number of turning points
Proposed algorithm	622.32	10.54	45
A* search algorithm	638.12	18.20	177

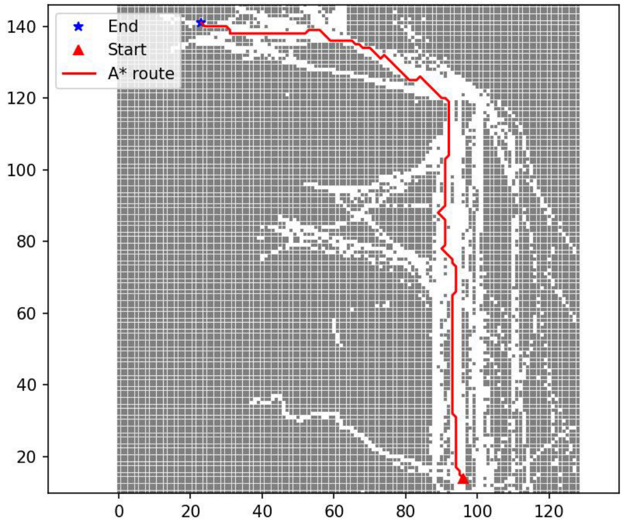

FIGURE 15. The result of the contrast algorithm.

V. CONCLUSION

This paper proposes an automatic planning algorithm that combines the ACA and the A* search algorithm for the shortest route. Firstly, a low-precision environment model is constructed based on the AIS data, and the grid range of the shortest route is determined using the ACA to ensure the global optimum and reduce the calculation nodes. Secondly, the A* search algorithm determines the shortest route under a more refined environment model. Thirdly, the path's key points are extracted and optimized using the Bézier curve method to obtain the shortest route that meets the actual requirements.

The experiments show that the algorithm proposed in this paper can reduce the computation nodes and obtain the shortest route that meets the safety requirements. The proposed algorithm has strong optimization capabilities and can provide meaningful guidance for shortest-path planning for intelligent ships or even unmanned ships in unknown obstacle environments, promoting the intellectual development of shipping.

This study proposes a method with some limitations, and we can still enhance the planning performance by following directions. Firstly, a high-precision environment model is required to compute the shortest and safe route. Building a high-precision environment model requires massive AIS data to prevent blind zones. Secondly, additional information, such as ENC data, needs to be introduced when building the environmental model to determine navigable areas. Thirdly, when the shortest distance is recalculated using the A* search algorithm, the heuristic function must be re-optimized, and the traditional 8-neighborhood A* search algorithm is used in this study. For better accuracy, 16-neighborhood and even 32-neighborhood A* search algorithm should be introduced.

REFERENCES

[1] J. Yang, L. Wu, and J. Zheng, "Multi-objective weather routing algorithm for ships: The perspective of shipping company's navigation strategy," *J. Mar. Sci. Eng.*, vol. 10, no. 9, p. 1212, 2022.

[2] J. Yu, G. Liu, J. Xu, Z. Zhao, Z. Chen, M. Yang, X. Wang, and Y. Bai, "A hybrid multi-target path planning algorithm for unmanned cruise ship in an unknown obstacle environment," *Sensors*, vol. 22, no. 7, p. 2429, Mar. 2022.

[3] L. Zhao, F. Wang, and Y. Bai, "Route planning for autonomous vessels based on improved artificial fish swarm algorithm," *Ships Offshore Struct.*, to be published.

[4] S. Guo, X. Zhang, Y. Zheng, and Y. Du, "An autonomous path planning model for unmanned ships based on deep reinforcement learning," *Sensors*, vol. 20, no. 2, p. 426, Jan. 2020.

[5] W. Zhao, H. Wang, J. Geng, W. Hu, Z. Zhang, and G. Zhang, "Multi-objective weather routing algorithm for ships based on hybrid particle swarm optimization," *J. Ocean Univ. China*, vol. 21, no. 1, pp. 28–38, Feb. 2022.

[6] S. Zhou, Z. Wu, and L. Ren, "Ship path planning based on buoy offset historical trajectory data," *J. Mar. Sci. Eng.*, vol. 10, no. 5, p. 674, May 2022.

[7] Z. Wang, S. J. Li, L. H. Zhang, and N. Li, "A method for automatic routing based on route binary tree," *Geomatics Inf. Sci. Wuhan Univ.*, vol. 35, no. 4, pp. 407–410, 2010.

[8] J. Lv and Z. N. Liu Wang, "Method for automatic ship routing based on route stack," *J. Comput. Appl.*, vol. 38, no.S1, pp. 16–19, 2018.

[9] T. Wang, L. H. Zhang, R. C. Peng, H. B. Cao, and L. J. Jiang, "A method for automatically generating the shortest distance route based on electronic navigational chart considering channel width," *Hydrographic Surveying Charting*, vol. 36, no. 3, pp. 29–31, 2016.

[10] S. Guo, X. Zhang, Y. Du, Y. Zheng, and Z. Cao, "Path planning of coastal ships based on optimized DQN reward function," *J. Mar. Sci. Eng.*, vol. 9, no. 2, p. 210, Feb. 2021.

[11] S. Broumi, A. Bakal, M. Talea, F. Smarandache, and L. Vladareanu, "Applying Dijkstra algorithm for solving neutrosophic shortest path problem," in *Proc. Int. Conf. Adv. Mech. Syst. (ICAMechS)*, Nov. 2016, pp. 412–416.

[12] K. Takashima, B. Mezaoui, and R. Shoji, "4 On the fuel saving operation for coastal merchant ships using weather routing," in *Marine Navigation and Safety of Sea Transportation*. Boca Raton, FL, USA: CRC Press, 2009, pp. 457–462.

[13] P. Krata and J. Szlapczynska, "Ship weather routing optimization with dynamic constraints based on reliable synchronous roll prediction," *Ocean Eng.*, vol. 150, pp. 124–137, Feb. 2018.

[14] X. Mou, "Research on the optimization of economical route for sea vessels based on dynamic programming algorithm," *Wuhan Univ. Technol.*, to be published.

[15] H. J. Wang and J. H. Xu Zhao, "Path planning of greenhouse robot based on potential field ant colony algorithm," *Jiangsu Agricult. Sci.*, vol. 45, p. 18, pp.222-225, 2017.

[16] L. Wang, Z. Zhang, Q. Zhu, and S. Ma, "Ship route planning based on double-cycling genetic algorithm considering ship maneuverability constraint," *IEEE Access*, vol. 8, pp. 190746–190759, 2020.

[17] P. Andersson and P. Ivehammar, "Dynamic route planning in the Baltic sea region—A cost-benefit analysis based on AIS data," *Maritime Econ. Logistics*, vol. 19, no. 4, pp. 631–649, 2017.

[18] L. Dong, J. Li, W. Xia, and Q. Yuan, "Double ant colony algorithm based on dynamic feedback for energy-saving route planning for ships," *Soft Comput.*, vol. 25, no. 7, pp. 5021–5035, Apr. 2021.

[19] H. Li, J. Liu, K. Wu, Z. Yang, R. W. Liu, and N. Xiong, "Spatio-temporal vessel trajectory clustering based on data mapping and density," *IEEE Access*, vol. 6, pp. 58939–58954, 2018.

[20] P. Han and X. Yang, "Big data-driven automatic generation of ship route planning in complex maritime environments," *Acta Oceanol. Sinica*, vol. 39, no. 8, pp. 113–120, Aug. 2020.

[21] S.-K. Zhang, G.-Y. Shi, Z.-J. Liu, Z.-W. Zhao, and Z.-L. Wu, "Data-driven based automatic maritime routing from massive AIS trajectories in the face of disparity," *Ocean Eng.*, vol. 155, pp. 240–250, May 2018.

[22] C. Liu, Q. Mao, X. Chu, and S. Xie, "An improved A-star algorithm considering water current, traffic separation and berthing for vessel path planning," *Appl. Sci.*, vol. 9, no. 6, p. 1057, Mar. 2019.

[23] R. Song, Y. Liu, and R. Bucknall, "Smoothed A* algorithm for practical unmanned surface vehicle path planning," *Appl. Oceans Res.*, vol. 83, pp. 9–20, 2019.

[24] M. Dorigo and L. M. Gambardella, "Ant colony system: A cooperative learning approach to the traveling salesman problem," *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 53–66, Aug. 1997.

[25] C.-B. Cheng and C.-P. Mao, "A modified ant colony system for solving the travelling salesman problem with time windows," *Math. Comput. Model.*, vol. 46, nos. 9–10, pp. 1225–1235, 2007.

[26] N. Makariye, "Towards shortest path computation using Dijkstra algorithm," in *Proc. Int. Conf. IoT Appl. (ICIOT)*, May 2017, pp. 1–3.

YANFEI ZHANG received the M.S. degree from the Shanghai Ship and Shipping Research Institute, in 2019. He is currently an Assistant Researcher with the Shanghai Ship and Shipping Research Institute. His research interests include ship route planning, autonomous ship navigation, intelligent ship data analysis, and artificial intelligence.

YIYAN WEN received the master's degree from the Jiangsu University of Science and Technology, in 2013. He is currently an Associate Professor with the Shanghai Institute of Shipping and Transportation Science. His research interests include ship speed optimization, autonomous ship navigation, and intelligent ship data analysis.

HAIYANG TU is currently a Professor with the Shanghai Institute of Shipping and Transportation Science. His research interests include ship speed optimization, autonomous ship navigation, and intelligent ship data analysis.