



## Dynamic anti-collision A-star algorithm for multi-ship encounter situations

Zhibo He <sup>a,b,c</sup>, Chenguang Liu <sup>a,c,e,\*</sup>, Xiumin Chu <sup>a,c</sup>, Rudy R. Negenborn <sup>d</sup>, Qing Wu <sup>a,b,c</sup>

<sup>a</sup> Intelligent Transport System Research Center, Wuhan University of Technology, Wuhan, China

<sup>b</sup> School of Transportation and Logistics Engineering, Wuhan University of Technology, Wuhan, China

<sup>c</sup> National Engineering Research Center for Water Transport Safety, Wuhan University of Technology, Wuhan, China

<sup>d</sup> Department of Maritime and Transport Technology, Delft University of Technology, Delft, The Netherlands

<sup>e</sup> Chongqing Research Institute, Wuhan University of Technology, Chongqing, China

### ARTICLE INFO

#### Keywords:

A-star algorithm  
Dynamic anti-collision  
COLREGS  
Ship domain  
Multi-ship encounter

### ABSTRACT

For the complex multi-ship encounter scenarios, this article proposes a dynamic collision avoidance path planning algorithm based on the A-star algorithm and ship navigation rules, namely Dynamic Anti-collision A-star (DAA-star) algorithm. A dynamic search mechanism of the DAA-star algorithm considering time factors is designed to enable the collision avoidance for situations with known moving obstacles. A quaternion ship domain is generated based on Automatic Identification System (AIS) data, and the navigation risk cost is calculated with the combination of the quaternion ship domain and potential field. The searching constraints conforming with the Regulations for Preventing Collision at Sea (COLREGS) rules are set for the DAA-star algorithm to guarantee the safety of collision avoidance. Meanwhile, the individual ship maneuverability constraints and maneuverability differences from ship to ship are both considered in the proposed DAA-star algorithm, which can solve the path planning problem with dynamic obstacles in multi-ship encounter scenarios. The simulation results show that, compared with the traditional A-star algorithm and dynamic A-star algorithm, the DAA-star algorithm can generate more reasonable dynamic and static obstacle avoidance paths in complex navigation scenarios in the trade-off between the navigation risk and economical efficiency.

## 1. Introduction

### 1.1. Background

In recent years, unmanned vehicles such as Unmanned Ground Vehicles (UGVs), Unmanned Aerial Vehicles (UAVs), and Unmanned Underwater Vehicles (UUVs) have received more and more attention (Bi, 2021). The rapid development of unmanned vehicle related technologies promotes the evolution of intelligent and unmanned ships (Felski and Zwolak, 2020; Zheng et al., 2017). The autonomous navigation control system of an unmanned ship mainly involves three parts: perception and cognition module, decision-making and path planning module, path following and implementation module (Yang et al., 2007). The International Maritime Organization (IMO) had formally proposed the concept of Maritime Autonomous Surface Ships (MASS) in 2017 (IMO, 2017). It formulated relevant regulations, which indicates that autonomous ships have become the inevitable development direction of ships in the future. Although many advances have been made in intelligent ship perception equipment such as Automatic Identification System (AIS) and Radar, ship collision accidents still occur, which could cause tremendous losses of lives and economics (Li

et al., 2021; Chauvin et al., 2013; Martins and Maturana, 2010). Actually, in several circumstances, the perception system has found obstacles on the ship tracking, however dangerous accidents such as collisions could still occur due to the lack of effective path planning methods. Therefore, when a ship is underway, especially in a complex multi-ship encounter scenario, the ship path planning is essential (Wang et al., 2021; Zhou et al., 2019). Moreover, it is also necessary to consider various factors such as the Regulations for Preventing Collision at Sea (COLREGS), the dynamic and static obstacles, the ship maneuverability constraints (Liu et al., 2018), the different ship intelligence levels (Huang et al., 2020), and the economy at the same time during path planning, which significantly increases the difficulty of ship path planning.

### 1.2. Related work

According to the type of obstacles, the ship path planning can be divided into global path planning, and local path planning (Larson et al., 2006). In this article, the obstacles are divided into known static, unknown, and dynamic obstacles. The known static obstacles refer

\* Corresponding author.

E-mail address: [liuchenguang@whut.edu.cn](mailto:liuchenguang@whut.edu.cn) (C. Liu).

to the static obstacles that have been detected, such as bridge piers, shores, anchored ships. The unknown obstacles refer to the obstacles that have not been detected, such as a small boat that suddenly appears. The unknown obstacles are not considered in the global path planning. The known dynamic obstacles refer to obstacles whose trajectories can be predicted, such as traffic lights, trains, and typical navigation ships. For the known dynamic obstacles, we can sample their trajectories according to time. Thus they can be considered as static obstacles within the sampled moment. For all the unknown obstacles, they can be regarded as dynamic obstacles. Therefore, during the actual path planning, obstacles can only be divided into static and dynamic ones. In general, the global path planning algorithm is utilized to avoid the static obstacles, and then the local path planning algorithm is utilized to avoid the dynamic obstacles.

Global path planning methods mainly include Bug, Dijkstra, A-star, Artificial Potential Field (APF), Particle Swarm Optimization (PSO). Bug algorithm (Lumelsky and Stepanov, 1986) is a simple obstacle avoidance algorithm whose idea is that the object walks along the obstacle's outline to bypass the obstacle after encountering the obstacle. This algorithm can always find a feasible path, but its efficiency is low and cannot handle complex scenarios. Dijkstra algorithm (Dijkstra et al., 1959) combines the breadth-first search algorithm and the concept of the heap. In the search process, it calculates the optimal path from the starting node to the target node and calculates the lowest cost path from the starting position to the target node. The A-star algorithm (Hart et al., 1968) introduces a heuristic function based on the Dijkstra algorithm and selects the best path by calculating the cost to reach the target, and the cost starts from the beginning. The APF algorithm (Khatib, 1986) sets a repulsive field for obstacles and a gravitational field for the target point. The object reaches the target point through the gradient descent algorithm. One of its disadvantages is that it may fall into a local deadlock and cannot reach the optimal solution. The PSO algorithm (Kennedy and Eberhart, 1995) assumes that each particle in the particle swarm represents a possible solution to a problem. The PSO algorithm is convergent and straightforward, and its speed is fast. However, there are problems with dimensionality disasters and local extreme values. All these algorithms have been applied to ship global path planning. Based on the traditional A-star algorithm, Liu et al. (2019), Singh et al. (2018) have taken into account collision risk, path length, and navigation rules to generate a safer, more feasible, and more economical path. In Chen et al. (2020), a fast-tracking algorithm is used to sum the impact of collision risk and generate the optimal global path.

The local path planning is to avoid moving or unknown obstacles nearby that are not considered in the global path planning. For the ship local path planning, the COLREGS proposed by IMO is often considered (mainly related to rules 13–17). Some scholars have adopted the method of dynamic real-time planning, that is, updating the map information with a certain frequency and then re-planning the path periodically (Jiang et al., 2014; Gibson et al., 2020). The three-dimensional reconstruction method is used to obtain the surrounding obstacle information in real time, and then the surrounding obstacles as static obstacles are regarded, and A-star algorithm is applied to re-plan the optimal path (Jiang et al., 2014). Gibson et al. (2020) propose a multi-agent time-based path planning method using A-star. Periodic updates of the generated path are calculated, utilizing the state feedback from the real world. One of the advantages of those methods is that they have simpler structures and easily realized features, and do not need to rely on other algorithms. Some scholars have also adopted this type of method in the local path planning of ships. In Naeem et al. (2012), obstacle information is detected and updated by continuously updating map information and re-plan the global path for current obstacles. The port side area of the ship obstacles is prohibited from being searched to comply with Rule 14 of the COLREGS rule. Nevertheless, its shortcomings are also apparent, i.e., each re-planning of the path

is accompanied by a certain amount of calculation. Re-plan method is easy to fall to an optimal local area and ignore the global optimal path.

Some scholars prefer planning a globally optimal path first and then avoiding dynamic obstacles by varying the speed. Khatib (1986) adopt a two-level planning method. A robot walking path is firstly planned according to the known static information in the environment, and then the robot is guided to run along the path and avoid dynamic obstacles by adjusting the speed of the robot. Fiorini and Shiller (1993) describe dynamic obstacles with the relative speed and position of the object. Thus, convert the dynamic path planning problem to the static path planning problem. To ensure the optimal path while reducing the risk of collision with dynamic obstacles. The advantage of these methods is that the object can always follow the shortest path (or the minimal risk path). But they have some disadvantages, for instance, a certain time cost increasing, and maneuverability limitations. It is difficult for some objects, especially ships, to continuously accelerate or decelerate or even stop during the driving process. Therefore, this type of method maybe not be suitable for complex waters.

In some scenarios, the modular concepts, i.e., dividing path planning to multiple modules, have also been considered. The global path planning uses the module based on an optimal algorithm. The local path planning uses a separate algorithm (Ajeil et al., 2020; Mobadersany et al., 2015). For instance, Mobadersany et al. (2015) proposed a hybrid path planning method combining a Dijkstra algorithm for global path planning and a fuzzy theory for local planning. The global path is divided into several small segments, with local collision avoidance within each small segment. At this stage, most ship path planning methods adopt multiple module methods. For the local path module, Lee et al. (2004) proposed an autonomous navigation algorithm for ships based on fuzzy logic based on the Virtual Force Field (VFF). The proposed navigation algorithm can comply with the COLREGS and cope with the uncertainties of the marine environment. However, this algorithm cannot deal with various complex navigation situations. In Lyu and Yin (2019), a real-time path planning method for autonomous ships in a complex environment is proposed. Specifically, the local path planning uses an improved APF algorithm, including a new and improved repulsion function and virtual force, which can solve collision avoidance problems for multiple dynamic ships and stationary obstacles in line with COLREGS rules. However, it is computationally intensive and cannot avoid the deadlock problem. Lee et al. (2019) propose a ship automatic collision avoidance and path generation algorithm based on Velocity Potential Field (VPF) method to avoid the deadlock problem. COLREGS rules are incorporated into the algorithm. At the same time, multiple-ship encounter situations in crowded waters are not taken into consideration. In conclusion, one of the advantages of the above methods is that it adopts modularization. Each module performs its duties, which can effectively solve the problem. The disadvantage is that the complexity has increased significantly. There are defects in the combined strategy of global path planning and local collision avoidance. The segmented method, i.e., the return path is generated after the partial collision avoidance is complete, will increase the cost of the journey. The re-planning method, i.e., re-planning after collision avoidance, the amount of calculation, will significantly increase. As shown in Table 1, there are contradictions among the increase of the calculation amount of path planning, the increase of factors considered in path planning, the real-time calculation, and the reliability. How to improve the path planning performance in consideration of COLREGS rules is a point worth researching.

Dynamic variants of the A-star algorithm are capable of collision avoidance in dynamic environments. The D-star algorithm is also called dynamic A-star search algorithm. Compared with the traditional A-star, the D-star algorithm and its variants change the cost function between nodes during the planning process (re-planning online) is equivalent to A-star re-planner (Zhu et al., 2021). Because the algorithm can plan a feasible path in an unknown environment, this method was used by National Aeronautics and Space Administration (NASA) to find a

**Table 1**  
Comparison between the proposed algorithm and other algorithms.

Article	Algorithm	Computational load	Single-ship or Multi-ships	COLREGS	Risk cost
Sang-Min Lee et al. (2004)	VFF	Big	Single	✓	✗
Wasif Naeem et al. (2012)	A-star	Small	Single	✓	✗
Chenguang Liu et al. (2019)	A-star	Small	Single	✗	✓
Man-Chun Lee et al. (2019)	VPF	Big	Single	✓	✓
Hongguang Lyu et al. (2019)	APF	Big	Multi	✓	✓
Our article	DAA-star	Small	Multi	✓	✓

path for the Mars rovers. However, the D-star algorithm has a high cost of the required memory. Moreover, it is more suited to flexible targets rather than poorly maneuverable targets like ships. The velocity obstacle (VO) method is more used in ship collision avoidance when the trajectory of the target ship is known (Huang et al., 2018, 2019). The trajectories of ships in a time-varying environment are computed using the VO method, which denotes the ship's velocities that would cause a collision with obstacles at some near-future time. The VO method can consider the kinematic constraints but cannot generate an optimal (shortest) path. If we consider path planning as a multi-constrained non-linear problem, then model predictive control (MPC) is a good approach (Murillo et al., 2018). MPC based planning methods can predict the trajectory of target ships and the impact of the behavior of own ship. However, at this stage the non-linear MPC does not meet the real-time requirements of the planning algorithm very well.

### 1.3. Major contribution

According to the current technical analysis on the problem of multi-vessel dynamic path planning, although several artificial intelligent-related path planning algorithms have successfully applied in dynamic planning, there are still possibilities that cannot find the optimal solution. By contrast, algorithms based on heuristic methods such as A-star-related algorithms do not usually have such problem (Zeng and Church, 2009; Chen et al., 2016). However, a traditional A-star algorithm does not consider sailing risk and dynamic obstacles. The improved A-star algorithms do not always consider the COLREGS rule for multiple ship encounter scenarios. In response to these problems, this article proposes the following contributions:

- Different from the reconstruction planning after updating the map, the moving obstacles with known trajectories are regarded as static obstacles and brought to an improved A-star algorithm for planning. By taking account into obstacle handling during the search process of A-star algorithm, the unification of global and local path planning is realized, and significantly the number of calculation decreases.
- By setting the risk cost based on ship domain and control cost, the algorithm-planned ship travel path is made safer, and the collision avoidance path avoiding other encountering ships is more in line with Rule 13–17 of the COLREGS.
- By sorting different ships' maneuverability, different priorities are set for multi-vessel path planning to solve the problem improve the safety level of multi-vessel encounter path planning.

### 1.4. Structure

This article is organized as follows. In Section 2, our dynamic anti-collision A-star (DAA-star) algorithm and risk cost model are introduced. In Section 3, dynamic obstacle avoidance algorithms based on single-ship and multi-ship are proposed. The simulation experiments are conducted in Section 4. In Section 5, the conclusions and future directions are presented.

## 2. Dynamic anti-collision A-star algorithm

A novel A-star algorithm for dynamic anti-collision of ships, i.e., DAA-star algorithm, is proposed in this section, and the principle and application of which will be introduced and discussed. First of all, the traditional A-star algorithm is introduced and analyzed, and the minimum cost path problem is analyzed. Then the dynamic coordination mechanism of DAA-star algorithm is introduced. Then the path planning risk cost function is set. Finally, combining COLREGS with A-star algorithm, the DAA-star algorithm is proposed.

---

### Algorithm 1 Traditional A-star algorithm

---

```

1: Mark  $P[\text{start}]$  as  $\text{openlist}$ .
2: while  $\text{openlist} \neq \text{empty}$  do
3:   Select the node  $P[i]$  from the  $\text{openlist}$  whose value of evaluation
      function  $F(P[i])$  is smallest.
4:   Mark  $P[i]$  as  $\text{closedlist}$ .
5:   if  $P[i] = P[\text{end}]$  then
6:     return "path is found".
7:   else
8:     Select the successor node  $P_i[j]$  around the node  $P[i]$ , and
       calculate  $F(P_i[j])$ .
9:     if  $P_i[j]$  belongs to  $\text{obstacle}$  or  $\text{closedlist}$  node then
10:      continue;
11:    end if
12:    Mark  $P_i[j]$  as  $\text{openlist}$ .
13:    if  $P_i[j]$  belongs to  $\text{openlist}$  and  $F(P_i[j]) < F(P_m[j])$  when
         $P[m]$  was marked as  $\text{closedlist}$  then
14:      Set parent node of  $P[j]$  as  $P[i]$ ,  $F(P[i]) = F(P_j[j])$ .
15:    end if
16:  end if
17: end while
18: return "the path cannot be found".

```

---

### 2.1. Traditional A-star algorithm

A-star algorithm was first proposed in Hart et al. (1968), which is used to find the minimum cost path from the starting point  $P[\text{start}]$  to the end point  $P[\text{end}]$ . The minimum cost can be distance, risk cost, etc. It is worth emphasizing that the A-star algorithm is a grid map-based algorithm. Ship planners generally obtain information about the location of obstacles in the surrounding environment by devices such as electronic charts and nautical radar, and then convert this information into a grid map. The start point and the end point with the obstacle information in the generated grid map are selected to run the A-star algorithm. The pseudo-code of the traditional A-star algorithm is shown in Algorithm 1, where  $P[i]$ ,  $P_i[j]$  represent the current node and the nodes around the node  $P[i]$ , respectively. Nodes in the  $\text{openlist}$  and nodes in the  $\text{closedlist}$  represent reachable and unreachable nodes, respectively. In the traditional A-star algorithm the evaluation function  $F(P[i])$  of  $P[i]$  is defined as Eq. (1):

$$F(P[i]) = G(P[i]) + H(P[i]) \quad (1)$$

where  $G(P[i])$  is the cost of the path from  $P[\text{start}]$  to  $P[i]$ ,  $H(P[i])$  is the cost of the cheapest path from  $P[i]$  to  $P[\text{end}]$ . It is worth noting

that  $H(P[i])$  has several heuristic function forms, such as Manhattan distance, Euclidean distance, diagonal distance, etc. Different heuristic functions have an effect on the search accuracy and rate of the A-star algorithm. In this article, Euclidean distance heuristic functions are used in order to unify the calculations.

## 2.2. Dynamic updating mechanism

The searching process of the A-star algorithm for static obstacles is similar to a dynamic window method (Seder and Petrovic, 2007), except that the window becomes eight points around node  $P[i]$ , and the information about these obstacles do not change over time, i.e., the obstacles are all static. The traditional A-star algorithm can only avoid static obstacles. Avoiding dynamic obstacles does not mean that the trajectory of object and the trajectories of the dynamic obstacles cannot be intersected, i.e., reaching the same location at different times is allowed. If the generated path is given the attribute of time, the collision avoiding with dynamic obstacles can be realized in time-space dimension. In the previous analysis, it is difficult for a ship to continuously accelerate or decelerate or even stop during navigation. Therefore it is important to generate a path that can avoid dynamic obstacles even if the ship is assumed to travel at a uniform speed. Considering that A-star algorithm is searching based on the current node  $P[i]$ , we only need to determine the time period  $T_c$  it takes to travel from the starting point  $P[start]$  to the current node  $P[i]$ . The core idea of the dynamic updating mechanism is to bring the current node  $P[i]$  and the starting point  $P[start]$  to a subroutine called "findparent", whose pseudo-code is shown in Algorithm 2.

**Algorithm 2** findparent( $P[i]$ ,  $P[start]$ )

```

1:  $T_c = 0$ 
2: while  $P[i] \neq P[start]$  do
3:   Find the parent node  $P[i - 1]$  of the  $P[i]$ .
4:    $T_c = T_c + \text{cost time}$  from  $P[i - 1]$  to  $P[i]$ 
5:    $P[i] = P[i - 1]$ 
6: end while
7: return  $T_c$ .
```

If the problem is faced with a constraint based on the number of movement steps, for instance the movement of a chess piece on the board, its time period can be determined in a way based on the number of moving steps. In the game of chess, the king can move to any adjoining square not attacked by one or more of the opponent's pieces, and this move step counting as a single move of the king. While in the movement of a ship, it takes different time when move to square along a straight or diagonal on which it stands because of a longer path when move to a diagonal square. Thus, on the basis of the step moved on the grid map, we calculate the actual length of each movement, and divide it by the ship's speed. A more accurate time-consuming  $T_c$  is obtained and applied to the calculations in this article. As shown in Fig. 1, moving from the starting point to the current node takes 4 intermediate nodes, the time period  $T_c$  is 5 with the grid counting method. Using the total distance of the grid to calculate the time (defined as the average speed of the object lateral or longitudinal movement of a grid divided by the unit of time),  $T_c$  is 6.2. If  $T_c$  is calculated according to the actual smooth trajectory, it can improve some precision.

The subroutine further obtains  $T_c$  of the current node by accumulating the time of each movement. It is worth noting that for the same node, the corresponding  $T_c$  will change when its parent node has changed and  $T_c$  may be the same for different nodes. After sampling the trajectory of the known dynamic obstacle  $n$  to get its  $T_c$  position  $D_n[T_c]$ , these dynamic obstacles can be regarded as the known static obstacles marked as *closelist*. So far, the paths of obstacles are only imported to the *closelist*, it is also necessary to eliminate the nodes of these dynamic obstacles after each search in order to avoid their trajectories instead

of their paths. In Figs. 2(a) and 2(b), the current nodes are (3,4) and (4,3) respectively. They have the same time period  $T_c = 2.4 (\sqrt{2} + 1)$ , and the target ship's position in  $T_c$  is (4,5). Thus, in Fig. 2(a), (4,5) is adjoining node of (3,4), but it is not in the search range of *openlist*. In  $T_c = 2.4$ , (4,5) is marked as *closelist*. In Fig. 2(b), the target ship moves from (4,5) to (5,4). Since (4,5) is no longer occupied by the target ship, it is eliminated from the *closelist*. And becomes a successor node of (5,4), which exists in the *openlist*. In Fig. 2(f), (6,3) is occupied by the target ship, therefore it is temporarily added to the *closelist* instead of the *openlist*.

It is worth noting that when adding or removing dynamic obstacle targets in the *closelist* at the end of each loop, it should be determined whether the coordinate point  $D_n[T_c]$  is already in the *openlist*. If the dynamic obstacle target has been in the *openlist*, it can be ignored; otherwise, it will cause the problem of repeated search. The pseudo-code of the improved dynamic A-star algorithm is shown in Algorithm 3, where  $P[start]$  is the current node in  $i$ th step.

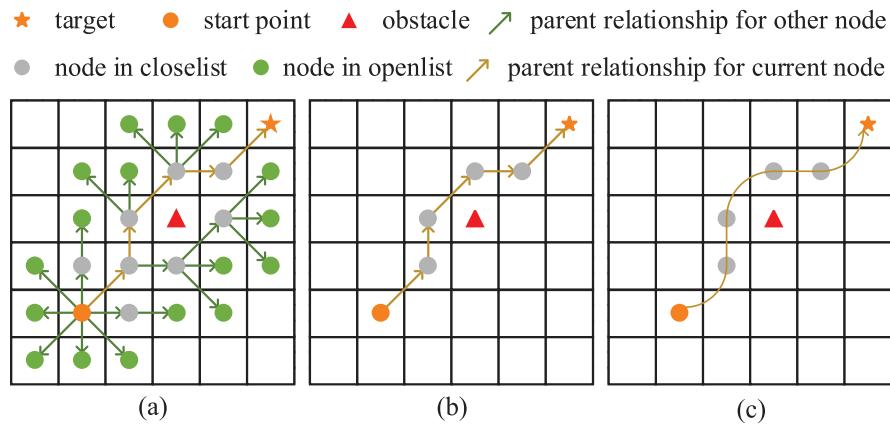
**Algorithm 3** Improved dynamic search A-star algorithm

```

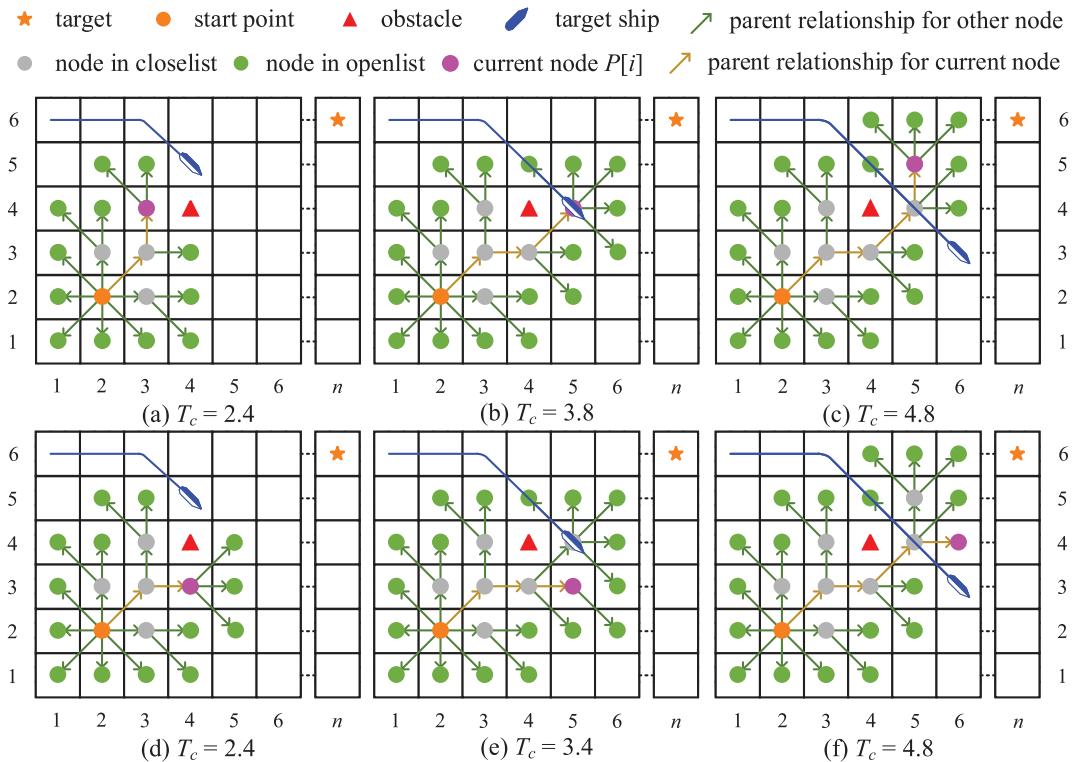
1: Mark  $P[start]$  as openlist.
2: while openlist ≠ empty do
3:   Select the node  $P[i]$  from the openlist whose value of evaluation
      function  $F(P[i])$  is smallest.
4:   Mark  $P[i]$  as closelist.
5:   if  $P[i] = P[\text{end}]$  then
6:     return "path is found".
7:   else
8:      $T_c = \text{findparent}(P[i], P[start])$ 
9:     Sampling the trajectory of the  $n$  known dynamic obstacles to
      get its  $T_c$  position  $D_n[T_c]$ .
10:    if  $D_n[T_c]$  does not belong to openlist then
11:      Mark  $D_n[T_c]$  as closelist.
12:    end if
13:    Select the successor node  $P_i[j]$  around the node  $P[i]$ , and
      calculate  $F(P_i[j])$ .
14:    if  $P_i[j]$  belongs to obstacle or closelist node then
15:      continue;
16:    end if
17:    Mark  $P_i[j]$  as openlist.
18:    if  $P_i[j]$  belongs to openlist and  $F(P_i[j]) < F(P_m[j])$  when
       $P[m]$  was marked as closelist then
19:      Set parent node of  $P[j]$  as  $P[i]$ ,  $F(P[i]) = F(P_i[j])$ .
20:    end if
21:    if  $D_n[T_c]$  does not belong to openlist then
22:      Remove  $D_n[T_c]$  from closelist.
23:    end if
24:  end if
25: end while
26: return "the path cannot be found".
```

## 2.3. Risk cost and operating cost modeling

The heuristic function is generally distance-related for the traditional A-star algorithm, so the A-star algorithm can always search for the shortest path, but the shortest path may not be suitable with concerning the safety factor. In Liu et al. (2019), the risk cost of obstacles is added to the A-star heuristic algorithm. By setting different weights between the distance and the risk cost, the shortest path is selected while ensuring safety. The generated path is more in line with the requirements of ship sailing. On the basis of Liu et al. (2019), this article will conduct the risk cost modeling from three points. The first step is to quantify the risk cost, the second step is to combine Rule 13–15 of the COLREGS, and the third step is to meet ship operability restrictions.



**Fig. 1.** The grid graph at a certain step in the search process of the A-star algorithm, (a) shows the application of other nodes in the search process, the arrow represents the parent-child relationship between the nodes from the parent node to the child node; (b) ignores other nodes and only keeps all parent node graphs between the current node and the starting point; (c) shows the smoothed trajectory graph.



**Fig. 2.** The search process of the A-star algorithm with the dynamic updating mechanism, the arrow represents the parent-child relationship between the nodes from the parent node to the child node.

### 2.3.1. Risk cost quantification

Ship domain is a generalization of a safe distance, and its introduction to maritime navigation comes from the observation that the safe distance is not the same in all direction. The ship domain is more and more used in path planning and ship collision avoidance (Tsou, 2016; Lazarowska, 2015). For the APF method, different repulsive force values are set for different hazard targets. However, for the grid map-based A-star algorithm, the setting of ship domain needs to be studied.

Considering that the main scene of this article is the Inland Waterway that could include Bridge area, multi-ship encounter and other obstacles scenarios. For the bridge area, we use the statistical method to study the ship domain in the bridge area reference in Hansen et al. (2013), Jinyu et al. (2021). Based on AIS data and preprocessing of AIS data, the ship with stable motion (mainly speed and course stability,

i.e. average speed or near constant speed passing through the target area without abnormal behavior) is selected as the target ship; The position of other obstacles (including sailing ships, piers and navigation marks) are calculated during the current trajectory time of the target ship. The multiple data are superimposed to obtain the ship domain of a single ship. According to the ship's length and speed, the data is classified and superimposed to obtain the ship domain of a specific ship type. In this article, the AIS data of bridge area in Wuhan Yangtze River water area are extracted, and the collection time is from September 19, 2019 to October 1, 2019. According to different ship speeds and ship length, the domain models of ships, piers and navigation aids are established. The flow chart is shown in Fig. 3.

Based on the AIS data analysis, the typical ship length is 100 m, the typical upstream median speed and downstream median speed of ship are 2.1 m/s and 3.6 m/s, which is selected as the setting of the ship

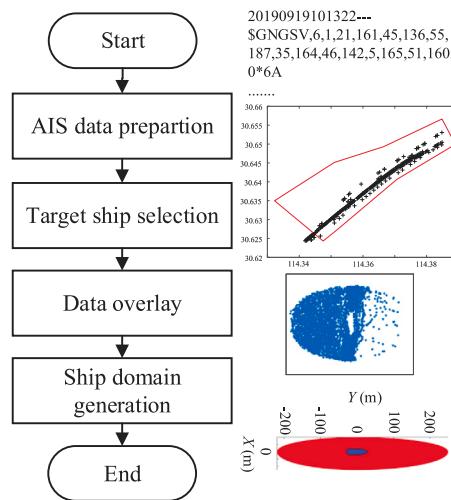


Fig. 3. Flow chart of ship domain generation.

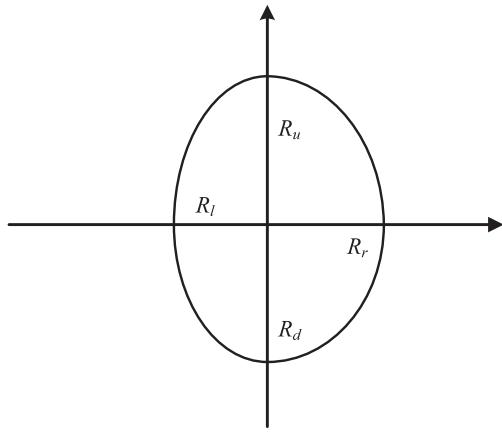


Fig. 4. Quaternion ship domain model.

domain. The generated four element ship domain model is defined as:

$$QSD = \{(x, y) \mid f(x, y, Q) \leq 1, Q = \{R_u, R_d, R_l, R_r\}\} \quad (2)$$

$$f(x, y) = \left( \frac{2x}{(1 + \text{sgn}(x)) R_u - (1 - \text{sgn}(x)) R_d} \right)^2 + \left( \frac{2y}{(1 + \text{sgn}(y)) R_l - (1 - \text{sgn}(y)) R_r} \right)^2 \quad (3)$$

$$\text{sgn}(x) = \begin{cases} 1, x \geq 0 \\ -1, x < 0 \end{cases} \quad (4)$$

where  $R_u$ ,  $R_d$ ,  $R_l$ ,  $R_r$  represent the length of the ship domain in four directions as shown in Fig. 4.

Then, we get the ship domain model with ship speed of 2.1 m/s and 3.6 m/s, and the domain of bridge piers and navigation marks,  $QSD_{sp1}$ ,  $QSD_{sp2}$ ,  $QSD_{pr}$ ,  $QSD_{nm}$  as shown in Fig. 5. To reduce the loss of navigation marks and ship collision, we set a safer domain of navigation marks.

$$QSD_{sp1} = \{(x, y) \mid f(x, y, Q) \leq 1, Q = \{170, 140, 50, 40\}\} \quad (5)$$

$$QSD_{sp2} = \{(x, y) \mid f(x, y, Q) \leq 1, Q = \{250, 220, 50, 40\}\} \quad (6)$$

$$QSD_{pr} = \{(x, y) \mid f(x, y, Q) \leq 1, Q = \{330, 220, 50, 50\}\} \quad (7)$$

$$QSD_{nm} = \{(x, y) \mid f(x, y, Q) \leq 1, Q = \{30, 20, 10, 10\}\} \quad (8)$$

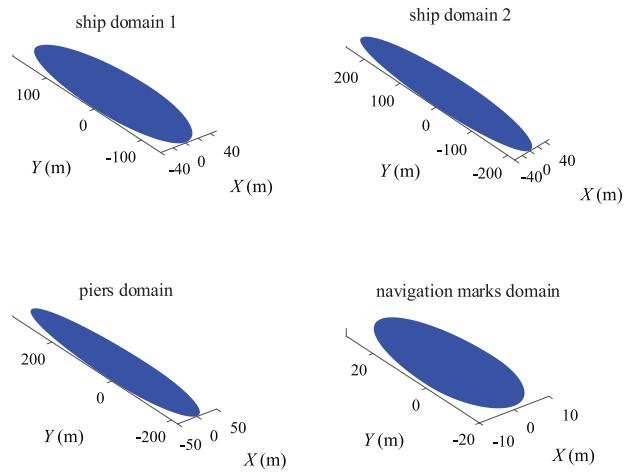


Fig. 5. Quaternion ship domain model in the bridge area.

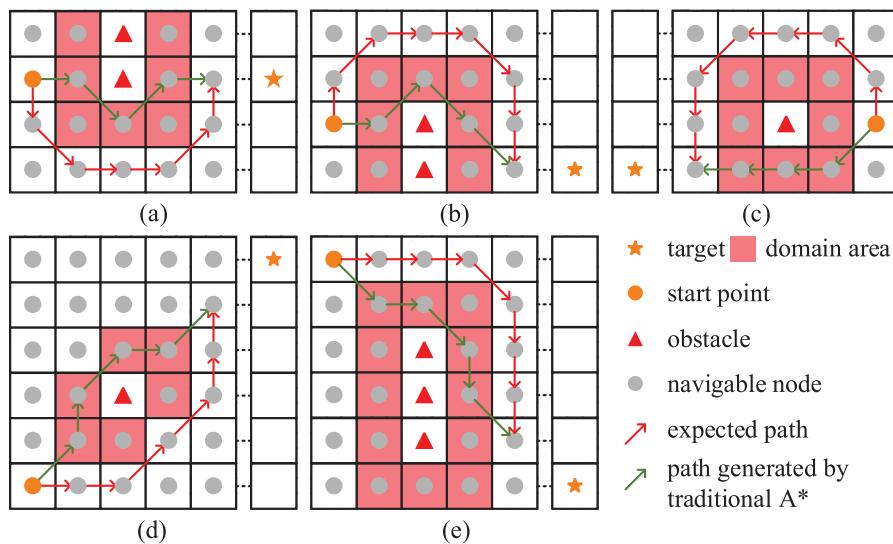
After obtaining the range of ship domain, we need to rasterize it to adapt to the A-star algorithm. Firstly, we select the appropriate grid length to sample the ship domain. The center point of the sampling grid in the ship domain is regarded as the grid of the ship domain, and the grid area of the ship domain is also regarded as obstacles. The A-star algorithm will ignore these nodes in the search process. As shown in Fig. 6, we can find that the A-star algorithm combined with the ship domain can generate a safer path.

Inspired by the APF method, we can choose a safer path while avoiding entering the ship domain. Here, we set up a potential field similar to APF, and the risk outside the ship domain is calculated based on the quaternion ship domain. Taking the pier as an example, the calculation of risk in  $P[i]$  outside the domain area of obstacle  $j$  is as follows:

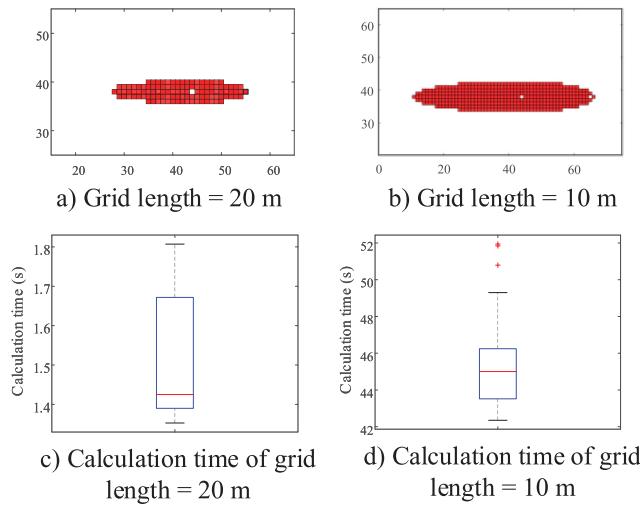
$$r_{ob,j}(P[i]) = \begin{cases} \left( \left( \frac{y_{o2b,i} \cos \theta - x_{o2b,i} \sin \theta}{R_u} \right)^2 + \left( \frac{x_{o2b,i} \cos \theta - y_{o2b,i} \sin \theta}{R_r} \right)^2 \right)^{-1}, & 0 < \theta \leq \pi/2 \\ \left( \left( \frac{y_{o2b,i} \cos \theta - x_{o2b,i} \sin \theta}{R_a} \right)^2 + \left( \frac{x_{o2b,i} \cos \theta - y_{o2b,i} \sin \theta}{R_r} \right)^2 \right)^{-1}, & \pi/2 < \theta \leq \pi \\ \left( \left( \frac{y_{o2b,i} \cos \theta - x_{o2b,i} \sin \theta}{R_a} \right)^2 + \left( \frac{x_{o2b,i} \cos \theta - y_{o2b,i} \sin \theta}{R_l} \right)^2 \right)^{-1}, & \pi < \theta \leq 3\pi/2 \\ \left( \left( \frac{y_{o2b,i} \cos \theta - x_{o2b,i} \sin \theta}{R_u} \right)^2 + \left( \frac{x_{o2b,i} \cos \theta - y_{o2b,i} \sin \theta}{R_l} \right)^2 \right)^{-1}, & 3\pi/2 < \theta \leq 2\pi \end{cases}, \quad (9)$$

where  $(x_{o2b,i}, y_{o2b,i})$  is the position of  $P[i]$  relative to the obstacle  $i$ ,  $\theta$  is the true azimuth of  $P[i]$  relative to the obstacle  $i$ .

Similarly, the choice of grid length also affects the final effect of path planning. Small grid length can increase the accuracy of rasterization in ship domain, but it will greatly increase the planning time of the A-star algorithm, as shown in Fig. 7. The map scale is a 1.5 km long and 1.5 km wide. When the grid length is 10 m (150\*150), the average time of planning 1.5 km path is 45 s, which is difficult to ensure the real-time requirements. When the grid length is 20 m (75\*75), the average time of planning a 1.5 km path is less than 2 s.



**Fig. 6.** The A-star algorithm combined with ship domain.



**Fig. 7.** Calculation difference of different grid length.

Choosing a larger grid scale will meet the real-time requirements of the algorithm, but will increase the risk of generate paths. The most basic grid map shall ensure that the quaternion ship domain is clearly differentiated in each quadrant. In other words, the grid length cannot exceed the minimum value of the ship's domain. While, choosing a small large grid scale can result in the algorithm taking too long to plan. The dynamic obstacle has already changed its position significantly before the algorithm has even generated a path. Therefore, it is worth noting to choose the appropriate grid length to ensure the accuracy or speed of path planning. We suggest that the calculation time of path planning algorithm should be less than 5 s. And the time also depends on the complexity of map. Based on the experience we have gained in our experiments, the total number of grids should preferably not be greater than 7500. Thus, considering the real-time requirements and low risk path, we give a suggested equation for the setting of the grid length as follows:

$$\sqrt{M_L M_W / N_g} \leq L_g \leq \min \{R_u, R_d, R_l, R_r\} \quad (10)$$

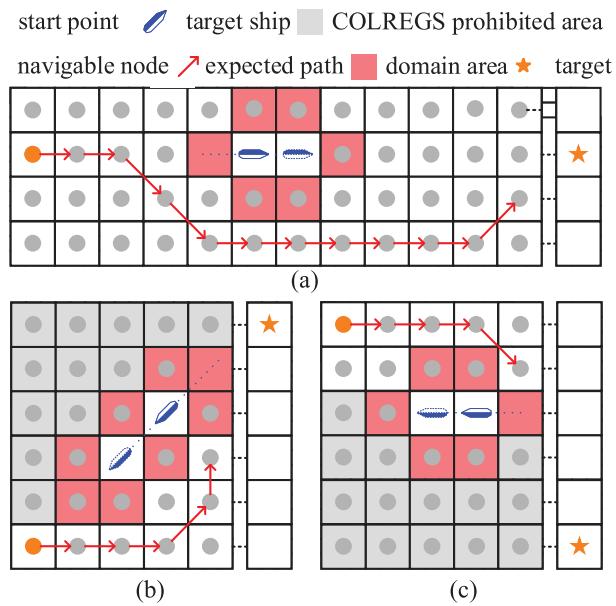
where  $L_g$  is the grid length,  $M_L$  is the map length,  $M_W$  is the map width,  $N_g$  is the suggestion maximum number of grids,  $R_u$ ,  $R_d$ ,  $R_l$ ,  $R_r$  represent the length of the ship domain in four directions.

### 2.3.2. COLREGS modeling

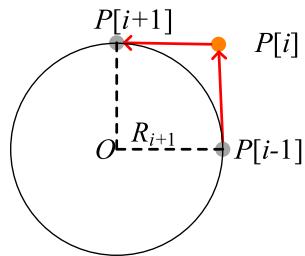
The encounter scenarios for any two vessels within the scope of the COLREGS can be divided into three categories: head-on, crossing, and overtaking. According to the Rule 13–15, for head-on situation, the two ships shall alter their course to starboard so that they shall pass on the port side of each other; for crossing situation, the ship which has target ship on her starboard side shall keep out of the way and shall avoid crossing ahead of target ship; for overtaking situation, any ship overtaking any other shall keep out of the way of the ship being overtaken. For each of the three encounter scenarios, the two ships have the action rules shown in Fig. 8. However, in a multi-ship encounter scenario, own ship will face a variety of encounter types at the same time. To make the paths planned by own algorithm follow the COLREGS as consistent as possible, we have introduced the recommendations in Rules 16–17 into own algorithm. Rule 16–17 give suggestion actions for give-way and stand-on ship. For emergency collision avoidance situations, the ship should avoid actions contrary to the recommendations of the COLREGS. For instance, we should avoid altering port side when target ship is coming from our port side. This is because if target ship complies with the COLREGS, our action will result in an increased risk of collision. Thus, for the target ship, we set the prohibited search area of A-star algorithm in various encounter scenarios, which can achieve the path production in line with COLREGS rules more easily, and reduce the search area of A-star algorithm at the same time. Considering that the interaction between ships does not keep connected all the time, it is not entirely sure whether other ships comply with the COLREGS rules or not during the procedure of anti-collision. In this article, ships are divided into two categories. One category is ships that can reliably exchange information with other ships, which can use the algorithm proposed in this article for path planning; the other category is ships that cannot reliably exchange information the trajectories of which are predicted based on their history trajectories. Because a ship cannot conduct acceleration, deceleration or turning action in a short time (Liu et al., 2017), it can be assumed that the ship's trajectory will not change abruptly, which can be predicted by Kalman filter or other means. In this article, we assume that all the ships except the own ship will adopt a negative attitude towards obstacle avoidance, i.e., they do not take the initiative to avoid obstacles, all obstacle avoidance measures are carried out only by the current ship using the DAA-star algorithm for path planning.

### 2.3.3. Maneuverability restriction modeling

The path planning algorithm should consider the actual ship motion limitations, e.g., the maneuverability restriction. If the path planned by



**Fig. 8.** Collision avoidance actions of three types of encounter scenes in rasterized maps, (a) shows the overtaking scene, (b) shows the oncoming scene, (c) shows the crossing scene.

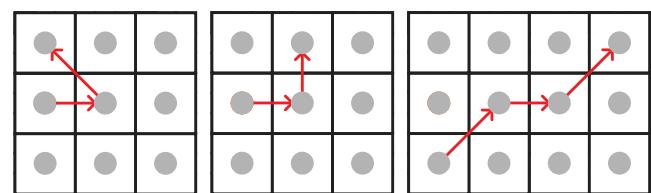


**Fig. 9.** The turning radius.

the A-star algorithm has a turning radius  $R_i$  less than the minimum turning radius  $R_0$  of the ship, it is difficult for the ship to follow the planned path, so maneuverability restrictions should be made to minimize the occurrence of such unreasonable paths. Referring to Liu et al. (2019), we calculate the turning radius  $R_{i+1}$  which from node  $P[i]$  to node  $P[i+1]$  by calculating the geometric relationship between  $P[i]$  and  $P[i-1]$ ,  $P[i+1]$ , as showed in Fig. 9. If the calculated turning radius  $R_{i+1}$  is greater than the ship's minimum turning radius  $R_0$ , it means that the path  $\{P[i-1], P[i], P[i+1]\}$  is unreasonable. In order to avoid this unreasonable path, and we set the turning risk cost of node  $P[i+1]$  as Eq. (11).

$$r_{tur}(P[i+1]) = \begin{cases} L_g \frac{R_0}{R_{i+1}}, & R_0 > R_{i+1} \\ 0, & R_0 > R_{i+1} \end{cases} \quad (11)$$

where  $L_g$  is the length of grid. At the same time, the ship cannot make too many maneuvers, such as the path in Fig. 10, which does not conform to the real driving performance of the ship. We can avoid these unreasonable paths by hard constraints and soft constraints, that is, by constraining the A-star algorithm to search the region or by increasing the cost. The former constraints are more rigorous, but there is a situation that the path cannot be found because of constraints are too strict. The latter constraints only needs to set a larger cost value to avoid these unreasonable paths. Therefore, it is necessary to increase the steering cost  $r_{str}$  whenever the path made a steer, so as to find the optimal path under the premise of the least number of turns.



**Fig. 10.** The unreasonable path.

### 3. Path planning

Based on the dynamic updating mechanism and risk-cost model established in Section 2, an improved dynamic obstacle avoidance A-star algorithm for ship dynamic path planning is proposed in this section. It is worth emphasizing that path planning in this context refers to the global path planning, where all the collision avoidance objects are known obstacles, i.e. known static obstacles and known dynamic obstacles.

#### 3.1. Single-ship dynamic path planning based on DAA-star algorithm

In this section, dynamic obstacles (target ships), risk costs of static obstacles, dynamic risk costs of dynamic obstacles, ship maneuverability, COLREGS, etc. are all considered in the dynamic path planning process for a single ship. We complemented the  $G(P[i])$  in Eq. (1), as shown in Eq. (12):

$$G(P[i]) = G_D(P[i]) + G_{tur}(P[i]) + G_{sta}(P[i]) + G_{dyn}(P[i]) + G_{str}(P[i]) \quad (12)$$

where  $G_D(P[i])$  is the distance cost from the starting point to the node  $P[i]$ ,  $G_{tur}(P[i])$  is the total turning risk cost of from the starting point to the node  $P[i]$ ,  $G_{sta}(P[i])$  is the total static risk cost from the starting point to the node  $P[i]$ ,  $G_{dyn}(P[i])$  is the total dynamic risk cost from the starting point to the node  $P[i]$  and  $G_{str}(P[i])$  is the total steering risk cost from the starting point to the node  $P[i]$ . When the algorithm is searching the node  $P[i]$  in  $T_c$ , the risk cost of the node  $P_i[j]$ ,  $r(P[i])$  is defined as Eq. (13):

$$r(P[i]) = r_{tur}(P[i]) + r_{sta}(P[i]) + r_{dyn}(P[i]) + r_{str}(P[i]) \quad (13)$$

where  $r_{tur}(P[i])$  is the turning risk cost,  $r_{sta}(P[i])$  is the static risk cost,  $r_{dyn}(P[i])$  is the dynamic risk cost and  $r_{dyn}(P[i])$  is the steering risk cost. The pseudo-code of the proposed single ship dynamic path planning algorithm is shown in Algorithm 4. In this algorithm,  $Obs_n[T_c]$  is a matrix of the domain area of dynamic obstacle  $n$  at moment  $T_c$ .  $P[start]$  is the current node in  $i$ th step.

#### 3.2. Multi-ships dynamic path planning based on DAA-star algorithm

Considering that not all ships have valid and reliable communication with each other, for instance, several ships do not turn on the AIS device even though it is the requirements of supervision department. Dynamic multi-vessel planning in this section does not imply path planning for all ships in the scenario. We consider some of the ships outside the master planning system (hereinafter called the ‘system’) keep their original paths, and the ships inside the system are dynamically planned based on the DAA-star algorithm.

For the ships inside the system, considering that the waterway could be occupied by the first-planned ship, the latter-planned ships require more maneuverability. Therefore, the ships in the system are based on their ship following ability index  $T$  (ship following ability refers to the ability of the ship) to change the direction of navigation. The following ability can be measured by the following ability index  $T$ . The smaller the value of  $T$  is, the better the following ability of the ship is, and vice

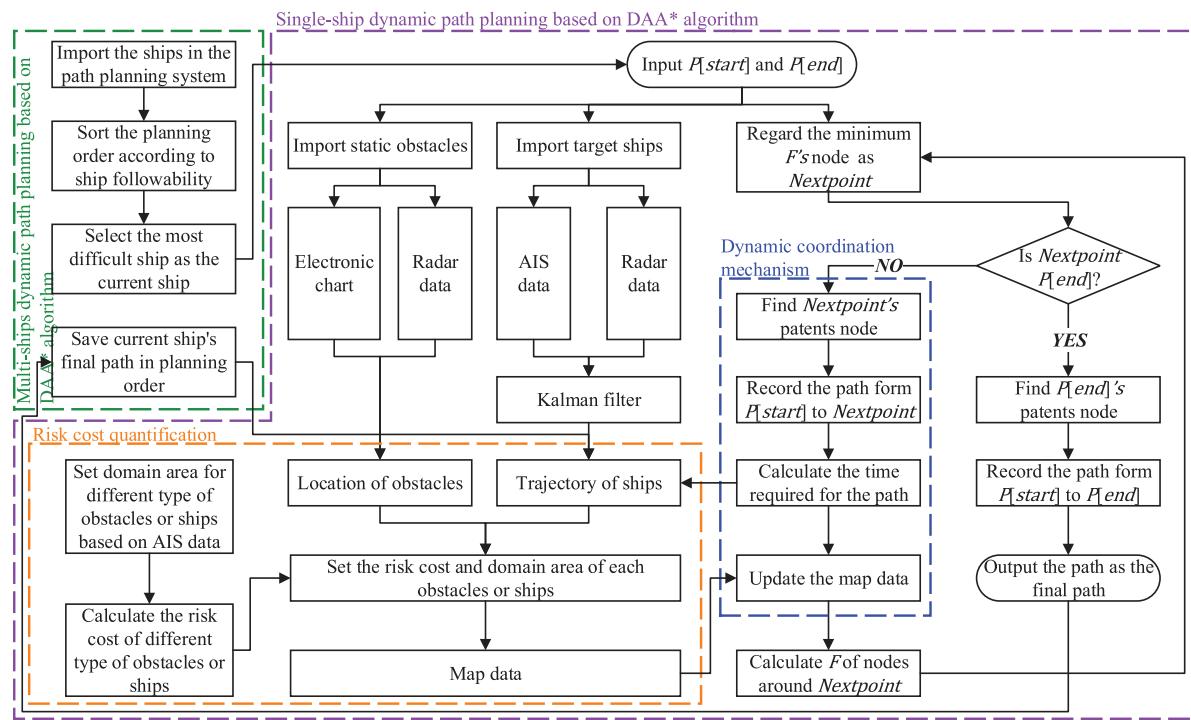


Fig. 11. Flow chart for dynamic anti-collision A-star algorithm for multi ships.

#### Algorithm 4 Dynamic anti-collision A-star algorithm for single ship.

```

1: Mark  $P[\text{start}]$  as  $\text{openlist}$ .
2: while  $\text{openlist} \neq \text{empty}$  do
3:   Select the node  $P[i]$  from the  $\text{openlist}$  whose value of evaluation
      function  $F(P[i])$  is smallest.
4:   Mark  $P[i]$  as  $\text{closelist}$ .
5:   if  $P[i] = P[\text{end}]$  then
6:     return "path is found".
7:   else
8:      $T_c = \text{findparent}(P[i], P[\text{start}])$ 
9:     Sampling the trajectory of the  $n$  known dynamic obstacles to
       get its  $T_c$  position  $D_n[T_c]$ , then generate the ship domain area
       and the COLREGS prohibited area, all this node defined as
        $\text{Obs}_n[T_c]$ .
10:    if  $\text{Obs}_n[T_c]$  does not belong to  $\text{openlist}$  then
11:      Mark  $\text{Obs}_n[T_c]$  as  $\text{closelist}$ .
12:    end if
13:    Select the successor node  $P_i[j]$  around the node  $P[i]$ , and
       calculate  $F(P_i[j])$ .
14:    if  $P_i[j]$  belongs to  $\text{obstacle}$  or  $\text{closelist}$  node then
15:      continue;
16:    end if
17:    Mark  $P_i[j]$  as  $\text{openlist}$ .
18:    if  $P_i[j]$  belongs to  $\text{openlist}$  and  $F(P_i[j]) < F(P_m[j])$  when
        $P[m]$  was marked as  $\text{closelist}$  then
19:      Set parent node of  $P[j]$  as  $P[i]$ ,  $F(P[i]) = F(P_i[j])$ .
20:    end if
21:    if  $\text{Obs}_n[T_c]$  does not belong to  $\text{openlist}$  then
22:      Remove  $\text{Obs}_n[T_c]$  from  $\text{closelist}$ .
23:    end if
24:  end if
25: end while
26: return "the path cannot be found".

```

versa. Therefore, the ship with a larger ship following ability index  $T$  will be given priority for path planning. After a ship in the system has planned a path, we regard the ship as a moving obstacle of a known path, and the path planning of other ships in the system will be carried out. The program pseudo-code is shown in Algorithm 5.

The structure  $\text{Ship}[ShipNum]$  in Algorithm 5 contains ship type information, starting point information, end point information, minimum turning radius information, ship size information, etc., which are used in the initial stage of path planning.  $MAP$  refers to the map information needed in the initial stage of algorithm planning, including channel information, stationary obstacle information, moving obstacle information, and their respective risk cost information. The flow chart of the entire multi-ship dynamic path planning is shown in Fig. 11.

#### Algorithm 5 Dynamic anti-collision A-star algorithm for multi ships

```

1: Read the number of ships that need route planning in the current
   system as  $ShipNum$ .
2: Sort these ships according to their ship following ability index  $T$ ,
   and store their information in the structure  $\text{Ship}[ShipNum]$ .
3:  $i = 1$ 
4: while  $i \neq ShipNum$  do
5:    $P[\text{start}] = \text{Ship}[i].Start$ 
6:    $P[\text{end}] = \text{Ship}[i].End$ 
7:    $PATH[ShipNum] = \text{DAA-star}(P[\text{start}], P[\text{end}], MAP)$ 
8:    $MAP = MAP + PATH[ShipNum]$ 
9:    $i = i + 1$ 
10: end while
11: return  $PATH$ 

```

#### 4. Case study

In order to assess the effectiveness of the DAA-star algorithm proposed in this article, simulation experiments are carried out in this section. The simulation platform is based on Intel i5 series 2.6 GHz processor with 16 GB memory, and the simulation software is MATLAB R2021a on Windows 10. Firstly, a path is generated by the DAA-star

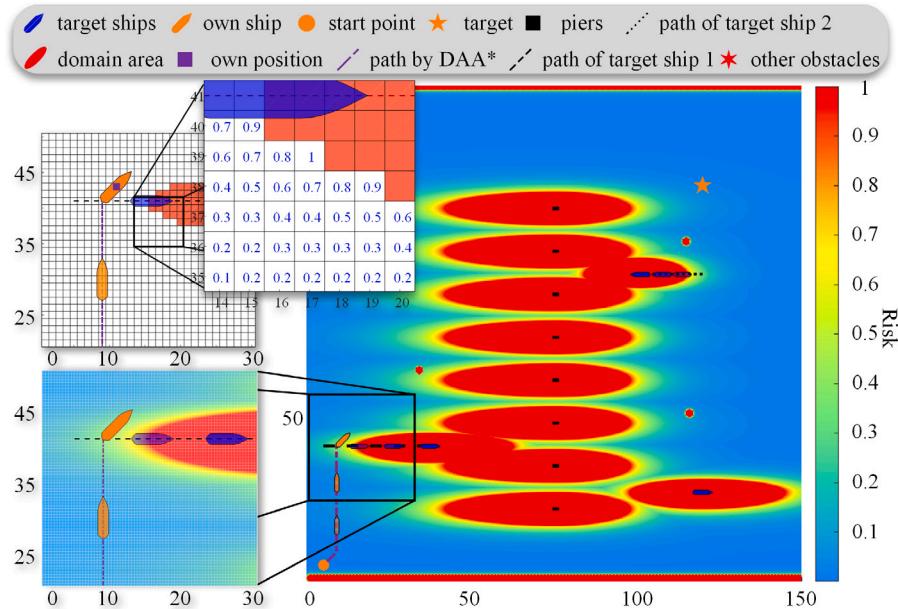


Fig. 12. Display of ship domain and risk cost on potential field map and grid map.

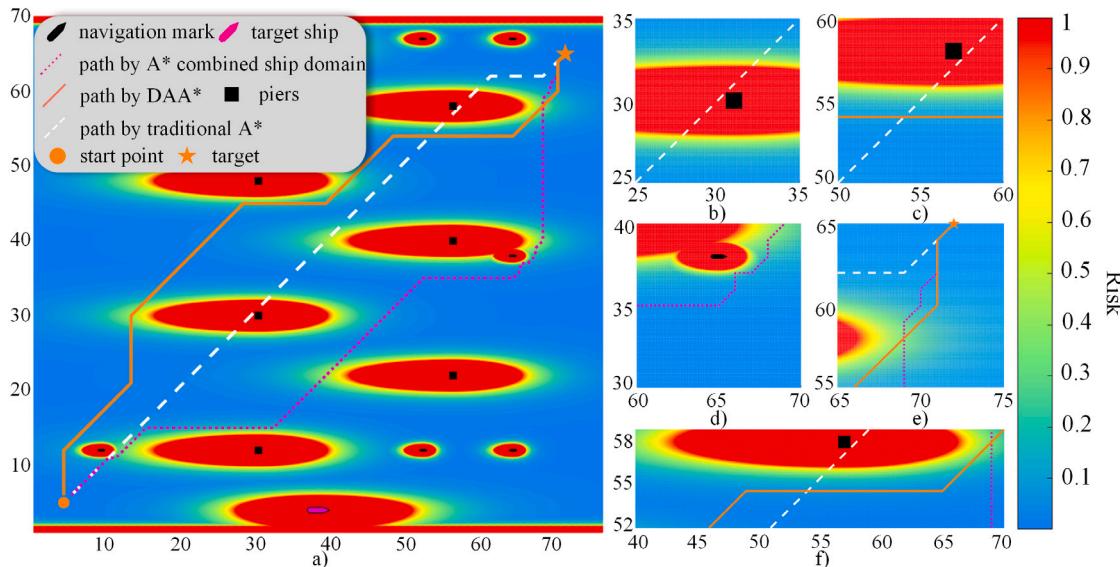


Fig. 13. Comparison of single ship static path planning of the traditional A-star algorithm, the A-star algorithm combined ship domain, and the DAA-star algorithm.

algorithm, and the differences between the DAA-star algorithm and the traditional A-star algorithm and the A-star algorithm combined ship domain are compared in the case of static obstacles. Then, the dynamic path planning for ship is performed and the planned path is evaluated using the proposed DAA-star algorithm, and the difference between the ‘Dynamic A-star algorithm’ proposed by Gibson et al. (2020) are compared in the case of dynamic obstacles. After that, the dynamic path planning for multi-ships is performed and the planned path is evaluated using the proposed dynamic multi-ship planning method based on the DAA-star algorithm. Finally, the proposed dynamic multi-ship planning method based on the DAA-star algorithm is used to generate the paths in a real scenario. It is worth noting that, for the convenience of display, the path planning results shown in this section are all displayed on the basis of the risk potential field, as shown in Fig. 12, rather than the grid map used by the traditional A-star algorithm, although the search process of the DAA-star algorithm is still based on grid map.

#### 4.1. Case 1: Static path planning for single ship

Here, we consider the limitations of ship maneuverability, and set some stationary obstacles. The starting point of the path planning is  $P[\text{start}] = (5, 5)$ , the end point is  $P[\text{end}] = (72, 65)$ , other parameters are shown in Table 2. The traditional A-star algorithm, the A-star algorithm combined ship domain, and the DAA-star algorithm are utilized, and the results are compared and shown in Fig. 13. The calculation time of each algorithm (50 cycles of testing), the length of the generated path, and the risk cost summation are all shown in Fig. 14.

The traditional A-star algorithm generates a short path (1,808 m) and a faster generation speed, but the generated path is too close to static obstacles, resulting in high collision risk (120). As shown in Fig. 13(b, c), the planned path almost fits the edge of the obstacle, which will cause a significant risk of collision in the real scene as shown in Fig. 15. Compared with the traditional A-star algorithm, the A-star algorithm combined ship domain produces a low risk path (20) with

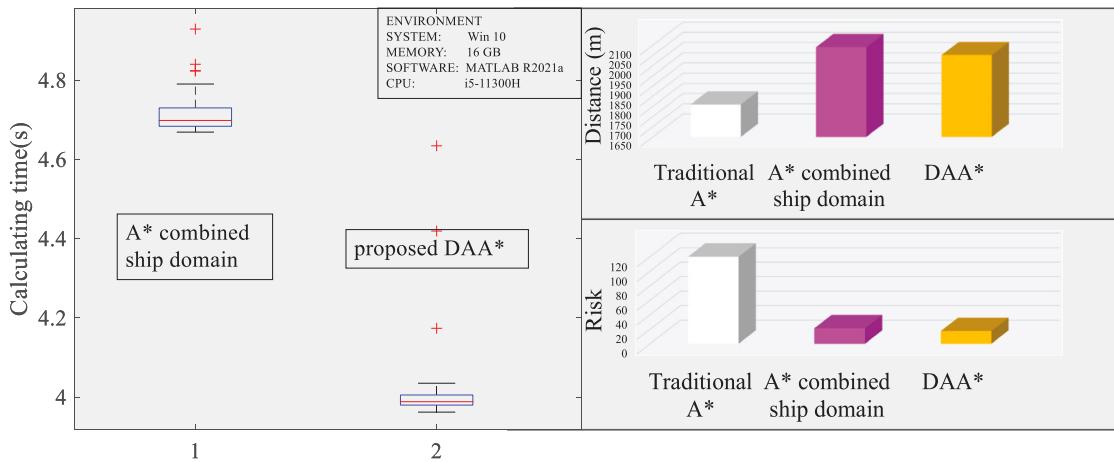


Fig. 14. Comparison of the calculation time and overall cost of the traditional A-star algorithm, the A-star algorithm combined ship domain, and the DAA-star algorithm.

Table 2

Parameter initialization case 1.

Parameters	Value
Horizontal grid number of map	77
Vertical grid number of map	70
Grid length $L_g$	20 m
Number of static obstacles	15

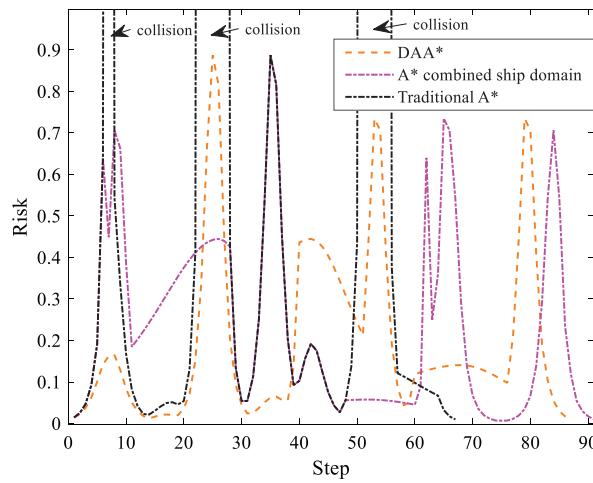


Fig. 15. Comparison of the collision risk of the whole process.

its length being 2,208 m, which has no essential difference with A-star algorithm. Meanwhile, this algorithm does not consider the ship maneuvering constraints, which could lead to generate unreasonable paths that a real ship cannot follow the path, as shown in Fig. 13(d, e). The DAA-star algorithm reduces the search time (15%) with the generated path as shown in Fig. 14. The increased risk of several collisions in Fig. 15 is due to the proximity to the edge of the piers domain, but according to the weighting of the algorithm, these risks are not sufficient for the ship to make a turn (a turn would add a greater cost). In summary, the DAA-star can generate a safe path that does not violate the ship domain, and has faster efficiency than the A-star algorithm combined ship domain search.

#### 4.2. Case 2: Dynamic path planning for single ships

In order to assess the effectiveness of the dynamic path planning with DAA-star algorithm, target ships are added in this case. Meanwhile, we test the dynamic A-star algorithm in Gibson et al. (2020)

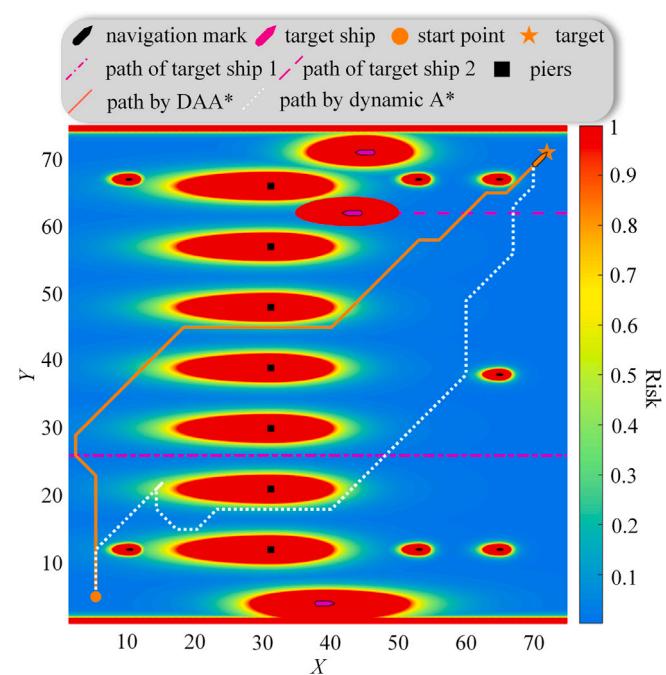
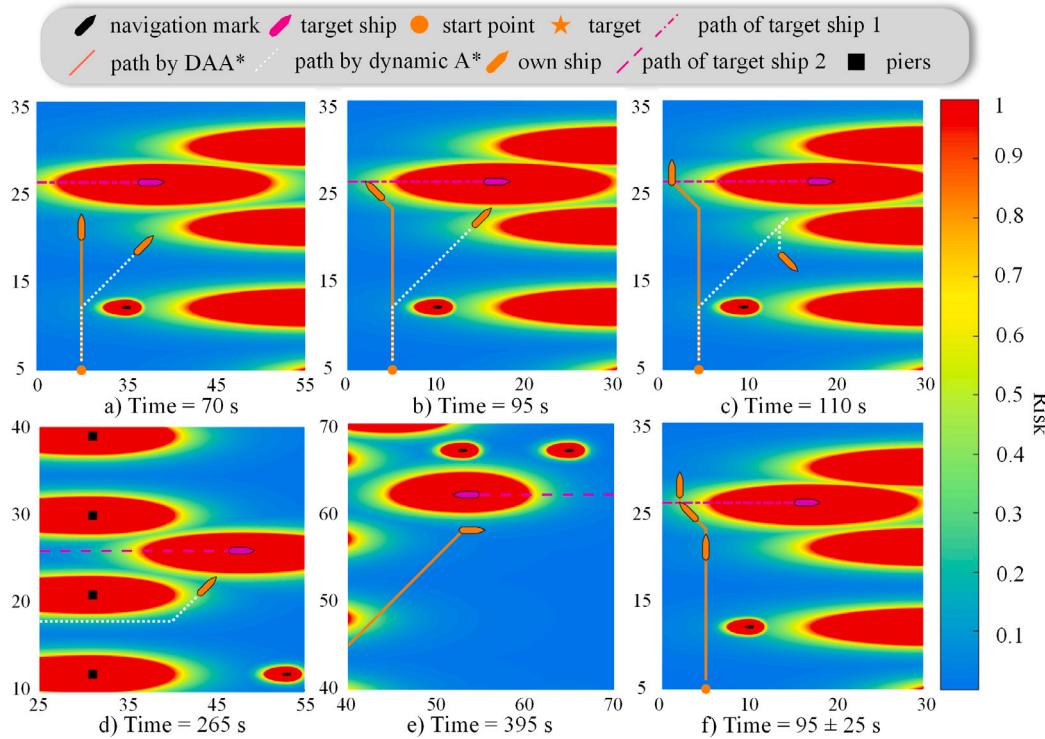


Fig. 16. Comparison of single ship dynamic path planning of the dynamic A-star algorithm and the DAA-star algorithm.

(combined with the ship domain set in this article) to generate a collision-free path. And the planning frequency is changed to 0.1 Hz. The starting point is  $P_2[\text{start}] = (5, 5)$ , the end point is  $P_2[\text{end}] = (72, 71)$  and other parameters are shown in Table 3. The generated path results are shown in Fig. 16, Fig. 17. Considering the trajectory prediction error of target ships, we add a safety margin to the safety domain of target ships, which makes the algorithm be able to plan a feasible dynamic collision avoidance path within a certain error range.

We can see that the path needs to be re-planned when the ship is about to collide at 95 s in 17(b). The re-planned path is shown in Fig. 17(c), and the path connection before and after planning does not meet the operational constraints. When the ship continues to travel to the moment of 265 s when the collision may occur again, the collision avoidance path cannot be found at this time (unless the original path returns or berths and waits) as shown in Fig. 17(d). It can be seen that the re-plan method is easy to fall into the local optimal and the generated path does not meet the operational constraints and other problems. Meanwhile, it needs to continue planning, the amount of its



**Fig. 17.** Comparison of single ship dynamic path planning of the dynamic A-star algorithm and the DAA-star algorithm.

**Table 3**  
Parameter initialization case 2.

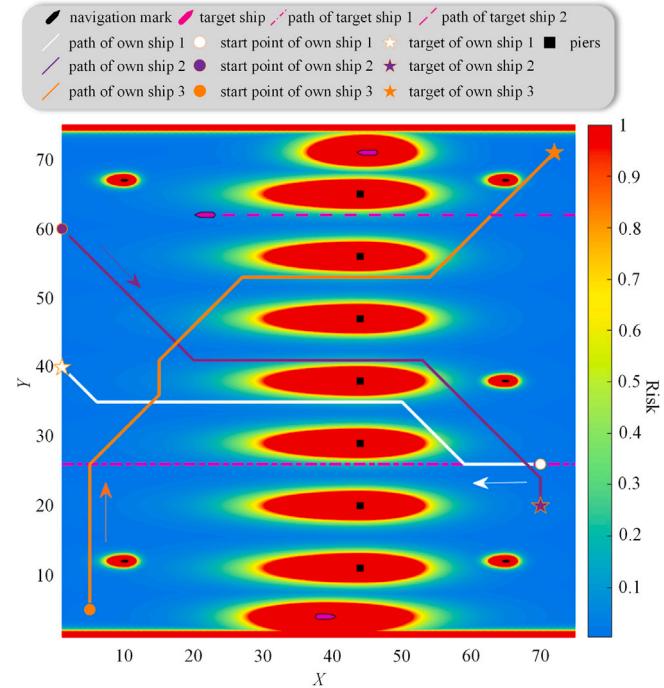
Parameters	Value	Parameters	Value
Horizontal grid number of map	75	Speed of target ship 1	3.6 m/s
Vertical grid number of map	75	Start point of target ship 1	(0,26)
Grid length	20 m	Course of target ship 1	90 deg
Number of static obstacles	16	Speed of target ship 2	2.1 m/s
Safety margin	1.5	Start point of target ship 2	(95,62)
Speed of Own ship	4.5 m/s	Course of target ship 2	270 deg

calculation is large. The DAA-star algorithm can consider the movement of obstacles in the planning process, and the generated path can effectively avoid collision with moving obstacles. Moreover, it will not produce singularity due to re-plan, generate a global optimal path, and will not fall into the local optimal. At the same time, considering the maneuverability delay of the ship, the planning time of the algorithm and the trajectory prediction error of moving obstacles, we test the mobile collision avoidance ability of the algorithm when the maneuvering delay and the maneuvering advance is 25 s (arriving at a node of the trajectory in advance or 25 s in advance). The results show that our algorithm can plan a dynamic collision avoidance path with safety redundancy.

#### 4.3. Case 3: Dynamic path planning for multi-ships

In order to assess the effectiveness of the algorithm for the multi-ship dynamic path planning, other ships are added in this case. We add ships with different speeds and choose the planning order according to their speeds. All parameters are shown in Table 4.

First of all, we choose ship 1 with a speed of 2.1 m/s as the priority planning ship, because its speed is too slow and its maneuverability is poor. After the feasible path is planned in the current dynamic scene, we sample the path of ship 1 and add it to the dynamic obstacles as the target ship with known trajectory. Then we plan the path of ship



**Fig. 18.** Dynamic path planning for multi-ships based on DAA-star algorithm.

2 with a speed of 5 m/s. Multi-ship path planning has been completed by multiple cycles and the results are shown in Fig. 18. The results show that the planned multi-ships paths do not conflict with each other and can avoid collision with other ships in the planning system while avoiding the static and dynamic obstacles.

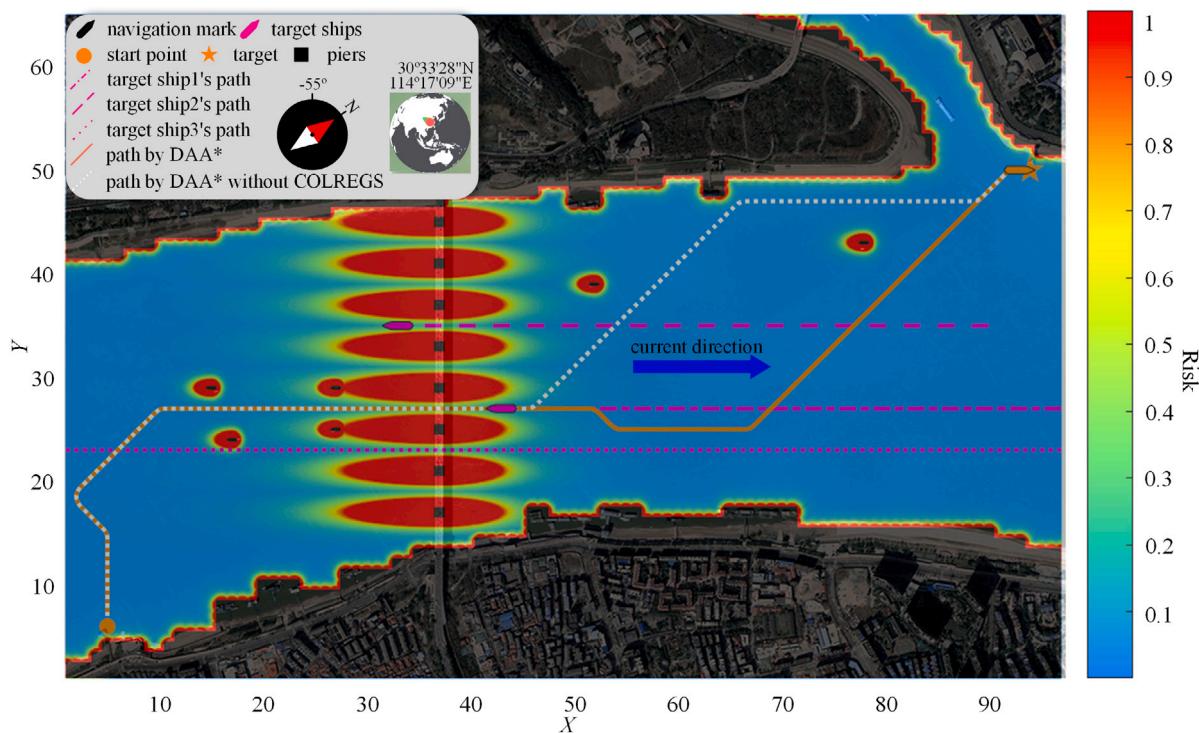


Fig. 19. Path planning by DAA-star in actual navigation.

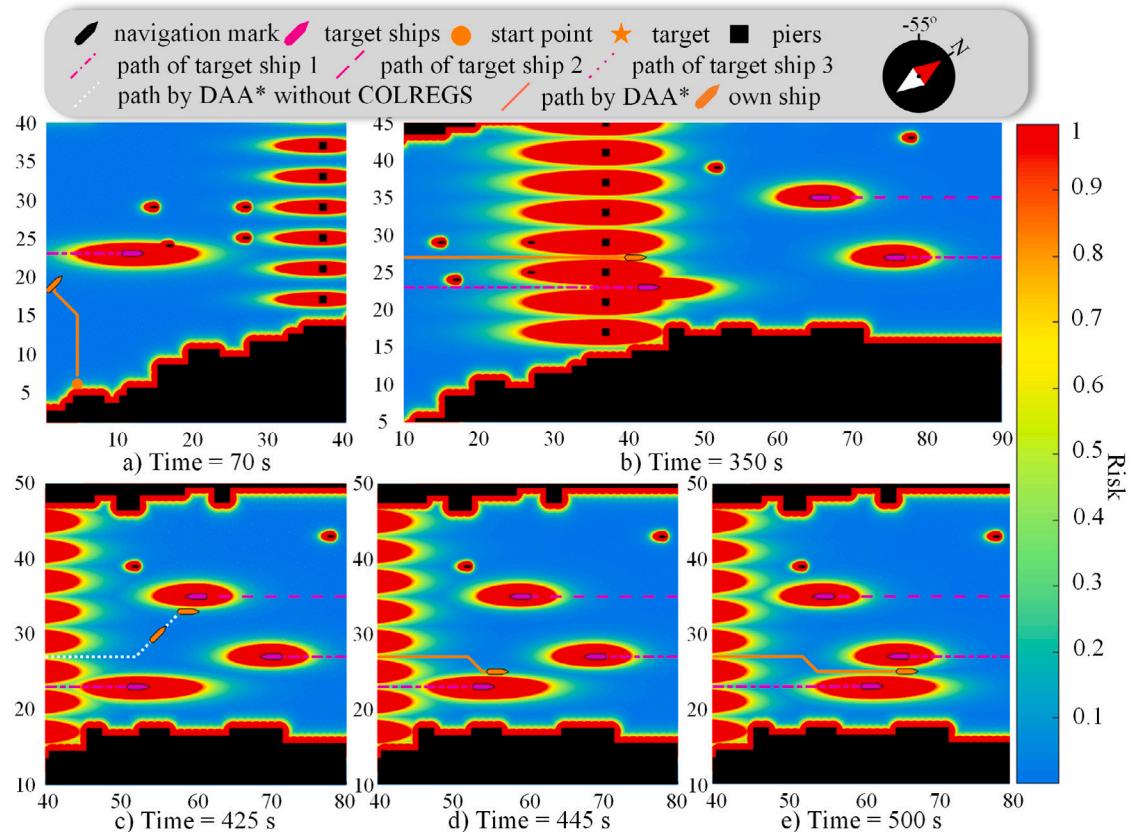


Fig. 20. Path planning by DAA-star in actual navigation.



**Fig. 21.** Satellite map of Case 4.

**Table 4**  
Parameter initialization case 3.

Parameters	Value	Parameters	Value
Horizontal grid number of map	75	Course of target ship 2	270 deg
Vertical grid number of map	75	Speed of own ship 1	2.1 m/s
Grid length	20 m	Start point of own ship 1	(70,26)
Number of static obstacles	14	Target of own ship 1	(1,40)
Safety margin	1.2	Speed of own ship 2	3.6 m/s
Speed of target ship 1	3.6 m/s	Start point of own ship 2	(1,60)
Start point of target ship 1	(0,26)	Target of own ship 2	(70,20)
Course of target ship 1	90 deg	Start point of own ship 3	(5,5)
Speed of target ship 2	2.1 m/s	Target of own ship 3	(72,71)
Start point of target ship 2	(95,62)	Speed of own ship 3	5.0 m/s

#### 4.4. Case 4: Dynamic path planning in real scene

In order to assess the effectiveness of the proposed DAA-star algorithm in this article, this case is based on the simulation of a Yangtze River waterway in Wuhan. In this channel scenario, there are static obstacles such as bridge piers and navigation marks, which are arranged according to the satellite map and the actual scene. In addition, in order to assess the effectiveness of the algorithm for dynamic obstacle avoidance, the AIS data of several ships are collected and their trajectories are recorded as the dynamic obstacles in this simulation. The center longitude and latitude of the scene as selected as (30° 33' 28" N, 114° 17' 09" E). At the same time, in order to better show the channel effect, we rotate the actual scene clockwise by 55 degrees. The area is a 3 km long and 2 km wide rectangle, and the location of the piers refers to the real location of the piers of Wuhan Yangtze River Bridge. The start point is (5,6) and the target is (94,50) and other parameters are shown in Table 5. It is worth noting that the trajectory of the target ship set in this case is only to produce a meeting scene, and the actual Yangtze River waterway counterparts implement the separation system, which will not produce a similar trajectory of the target ship 1. This case only shows the potential of DAA-star algorithm.

Considering that a feasible path could not be generated if all the ships comply the collision avoidance rules strictly in the inland waters,

**Table 5**  
Parameter initialization case 4.

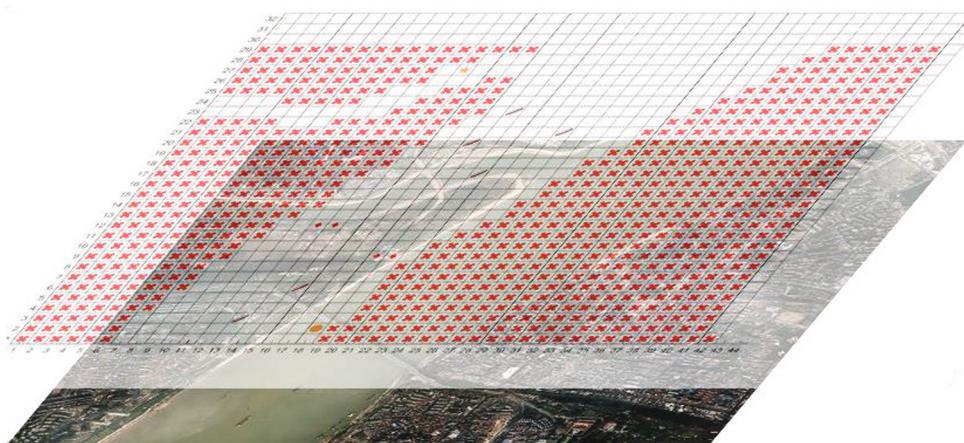
Parameters	Value	Parameters	Value
Horizontal grid number of map	100	Start point of target ship 2	(100,27)
Vertical grid number of map	66	Course of target ship 2	270 deg
Grid length	30 m	Speed of target ship 1	3.6 m/s
Number of static obstacles	14	Start point of target ship 1	(1,23)
Safety margin	1.5	Course of target ship 1	90 deg
Speed of target ship 1	2.1 m/s	Speed of own ship	4.5 m/s
Start point of target ship 1	(90,35)	Start point of own ship	(5,6)
Course of target ship 1	270 deg	Target of own ship	(94,50)
Speed of target ship 2	2.1 m/s		

we generate a path without considering the COLREGS rules in the actual scenario planning. The results are shown as Figs. 19 and 20.

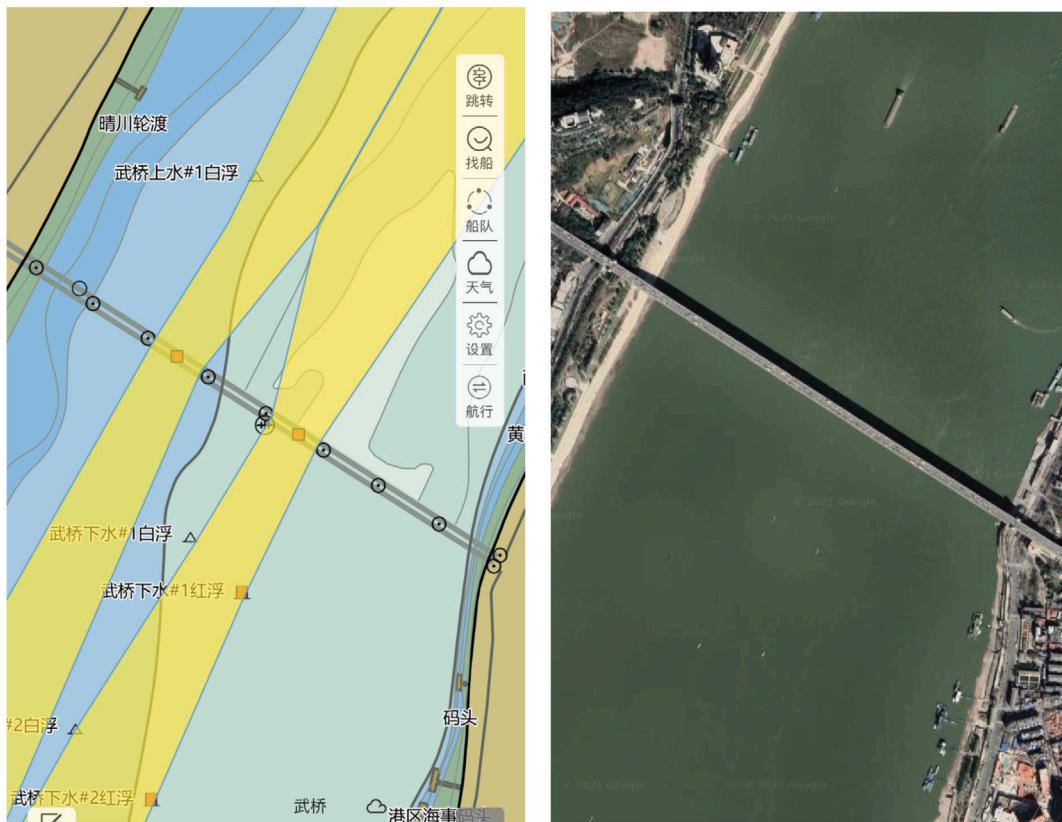
As shown in Fig. 20(a), the generate path starts to avoid collision when it has a certain safe distance. As shown in Fig. 20(b), at the moment of 350 s, the own ship and target ship 1 are facing the encounter scenario. At this time, the DAA-star algorithm considering COLREGs rules generates a path that conforms to the rules, as shown in Fig. 20(d, e), while the DAA-star algorithm without considering rules generates a path, as shown in Fig. 20(c). The results show that the DAA-star algorithm can generate a global optimal path in the real scene, which can realize the collision avoidance of dynamic obstacles with known trajectory.

#### 5. Conclusions and future research

In this article, a DAA-star algorithm is proposed to overcome the several shortcomings of traditional A-star algorithms, such as the inability to achieve dynamic planning, the high risk of generating paths, and the inability to combine COLREGS. The dynamic search principle of A-star algorithm is analyzed, and the mechanism of dynamic A-star algorithm considering the time factor is designed. The risk cost of different obstacles are calculated by the quaternion ship domain model, and the ship steerability limit is considered. The four cases' simulation results show that the proposed DAA-star algorithm can avoid dynamic



**Fig. 22.** Grid map based on satellite map. In which, the white grids mean the navigable grids, the red grids stand for banks. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



**Fig. 23.** Electronic map and Google Map in Case 4.

obstacles with known trajectories, and the collision avoidance paths can follow COLREGS. For known obstacles, the expected paths can be taken for obstacle avoidance, which significantly reduces the risk of collision. Because of the algorithm's low computing time, collision avoidance can be used for dynamic obstacles by re-plan.

However, the combination of ship planning path and actual control is not considered in this article, and the exact time required to follow the path is not considered in combination with a dynamic ship model. And the pose of the own ship model is not taken into consideration when expanding the domain. In addition, future research will also deal with the possible inconsistency of cooperation between ships with different intelligent levels.

#### CRediT authorship contribution statement

**Zhibo He:** Conceptualization, Methodology, Software, Validation, Editing. **Chenguang Liu:** Supervision, Conceptualization, Methodology, Resources. **Xiumin Chu:** Funding acquisition, Investigation, Validation. **Rudy R. Negenborn:** Software, Validation, Writing – reviewing. **Qing Wu:** Writing – reviewing and editing.

#### Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgment

This article is supported by National Natural Science Foundation of China (52001240), Natural Science Foundation of Chongqing, China (cstc2021jcyj-msxmX1220), Natural Science Foundation of Hubei Province (2020CFB307), Fundamental Research Funds for the Central Universities (203144003, 202444001, 213244001).

## Appendix

In Case 4, the real map is obtained from Google Maps as shown in Fig. 21. The bottom left corner of the map has a latitude and longitude of ( $30^{\circ} 33'28''\text{N}$ ,  $114^{\circ} 17'09''\text{E}$ ), which we regard as the center of the coordinate axis, with north being the positive  $y$ -axis direction and east being the positive  $x$ -axis direction. Dividing the map into multiple grids, according to the set grid length, the conversion of latitude and longitude coordinates to flat coordinates is generally done using methods such as the Mercator projection, but for scenes that do not span much latitude and longitude, some simpler methods can be used.

$$\begin{aligned}x_i &= C(Lat_i - Lat_0)/L_g \\y_i &= R(Lon_i - Lon_0)/L_g \\C &= R \cos(Lon_0) \\R &= 40076000 \text{ m}\end{aligned}\quad (14)$$

where  $R$  is the average circumference of the Earth,  $C$  is the circumference of current longitude,  $(Lat_0, Lon_0)$  stands for the bottom left corner of the map,  $L_g$  is the grid length. The initial grid map is generated by computer vision techniques (mainly based on connectivity domain detection) that identify the attributes (channels or banks) in each grid, as shown in Fig. 22.

The position of piers and navigation marks are obtained by the electronic map and Google Map as shown in Fig. 23.

## References

- Ajeil, F.H., Ibraheem, I.K., Sahib, M.A., Humaidi, A.J., 2020. Multi-objective path planning of an autonomous mobile robot using hybrid PSO-MFB optimization algorithm. *Appl. Soft Comput.* 89, 106076.
- Bi, X., 2021. Overview of autonomous unmanned systems. In: *Environmental Perception Technology for Unmanned Systems*. Springer, pp. 1–15.
- Chauvin, C., Lardjane, S., Morel, G., Clostermann, J.-P., Langard, B., 2013. Human and organisational factors in maritime accidents: Analysis of collisions at sea using the HFACS. *Accid. Anal. Prev.* 59, 26–37.
- Chen, P., Huang, Y., Papadimitriou, E., Mou, J., van Gelder, P., 2020. Global path planning for autonomous ship: A hybrid approach of fast marching square and velocity obstacles methods. *Ocean Eng.* 214, 107793.
- Chen, L., Negenborn, R.R., Lodewijks, G., 2016. Path planning for autonomous inland vessels using A\* BG. In: *International Conference on Computational Logistics*. Springer, pp. 65–79.
- Dijkstra, E.W., et al., 1959. A note on two problems in connexion with graphs. *Numer. Math.* 1 (1), 269–271.
- Felski, A., Zwolak, K., 2020. The ocean-going autonomous ship—Challenges and threats. *J. Marine Sci Eng* 8 (1), 41.
- Fiorini, P., Shiller, Z., 1993. Motion planning in dynamic environments using the relative velocity paradigm. In: [1993] *Proceedings IEEE International Conference on Robotics and Automation*. IEEE, pp. 560–565.
- Gibson, J., Schuler, T., McGuire, L., Lofaro, D.M., Sofge, D., 2020. Swarm and multi-agent time-based A\* path planning for LTA3 Systems. *Unmanned Syst.*
- Hansen, M.G., Jensen, T.K., Lehn-Schiøler, T., Melchior, K., Rasmussen, F.M.I., Enemark, F., 2013. Empirical ship domain based on AIS data. *The J. Navigation* 66 (6), 931–940.
- Hart, P.E., Nilsson, N.J., Raphael, B., 1968. A formal basis for the heuristic determination of minimum cost paths. *IEEE Trans. Syst. Sci. Cybern.* 4 (2), 100–107.
- Huang, Y., Chen, L., Negenborn, R.R., van Gelder, P., 2020. A ship collision avoidance system for human-machine cooperation during collision avoidance. *Ocean Eng.* 217, 107913.
- Huang, Y., Chen, L., Van Gelder, P., 2019. Generalized velocity obstacle algorithm for preventing ship collisions at sea. *Ocean Eng.* 173, 142–156.
- Huang, Y., Van Gelder, P., Wen, Y., 2018. Velocity obstacle algorithms for collision prevention at sea. *Ocean Eng.* 151, 308–321.
- IMO, 2017. Maritime safety committee (MSC), 98th session. [Online; accessed 25-November-2021], <https://www.imo.org/en/MediaCentre/MeetingSummaries/Pages/MSC-98th-session.aspx>.
- Jiang, S., Hong, Z., Zhang, Y., Han, Y., Zhou, R., Shen, B., 2014. Automatic path planning and navigation with stereo cameras. In: *2014 Third International Workshop on Earth Observation and Remote Sensing Applications (EORSA)*. IEEE, pp. 289–293.
- Jinyu, L., Lei, L., Xiumin, C., Wei, H., Xinglong, L., Cong, L., 2021. Automatic identification system data-driven model for analysis of ship domain near bridge-waters. *The Journal of Navigation* 1–22.
- Kennedy, J., Eberhart, R., 1995. Particle swarm optimization. In: *Proceedings of ICNN'95-International Conference on Neural Networks*. 4, IEEE, pp. 1942–1948.
- Khatib, O., 1986. Real-time obstacle avoidance for manipulators and mobile robots. In: *Autonomous Robot Vehicles*. Springer, pp. 396–404.
- Larson, J., Bruch, M., Ebken, J., 2006. Autonomous navigation and obstacle avoidance for unmanned surface vehicles. In: *Unmanned Systems Technology VIII*. 6230, International Society for Optics and Photonics, 623007.
- Lazarowska, A., 2015. Ship's trajectory planning for collision avoidance at sea based on ant colony optimisation. *The J. Navigation* 68 (2), 291–307.
- Lee, S.-M., Kwon, K.-Y., Joh, J., 2004. A fuzzy logic for autonomous navigation of marine vehicles satisfying COLREG guidelines. *Int. J. Control. Autom. Syst.* 2 (2), 171–181.
- Lee, M.-C., Nieh, C.-Y., Kuo, H.-C., Huang, J.-C., 2019. An automatic collision avoidance and route generating algorithm for ships based on field model. *J. Marine Sci Tech* 27 (2), 101–113.
- Li, L., Wu, D., Huang, Y., Yuan, Z.-M., 2021. A path planning strategy unified with a COLREGS collision avoidance function based on deep reinforcement learning and artificial potential field. *Appl. Ocean Res.* 113, 102759.
- Liu, J., Hekkenberg, R., Quadvlieg, F., Hopman, H., Zhao, B., 2017. An integrated empirical manoeuvring model for inland vessels. *Ocean Eng.* 137, 287–308.
- Liu, C., Mao, Q., Chu, X., Xie, S., 2019. An improved A-star algorithm considering water current, traffic separation and berthing for vessel path planning. *Appl. Sci.* 9 (6), 1057.
- Liu, C., Negenborn, R.R., Chu, X., Zheng, H., 2018. Predictive path following based on adaptive line-of-sight for underactuated autonomous surface vessels. *J. Marine Sci Technol* 23 (3), 483–494.
- Lumelsky, V., Stepanov, A., 1986. Dynamic path planning for a mobile automaton with limited information on the environment. *IEEE Trans. Automat. Control* 31 (11), 1058–1063.
- Lyu, H., Yin, Y., 2019. Colregs-constrained real-time path planning for autonomous ships using modified artificial potential fields. *The J. Navigation* 72 (3), 588–608.
- Martins, M.R., Maturana, M.C., 2010. Human error contribution in collision and grounding of oil tankers. *Risk Anal: An Int J* 30 (4), 674–698.
- Mobadersany, P., Khanmohammadi, S., Ghaemi, S., 2015. A fuzzy multi-stage path-planning method for a robot in a dynamic environment with unknown moving obstacles. *Robotica* 33 (9), 1869.
- Murillo, M., Sánchez, G., Genzelis, L., Giovannini, L., 2018. A real-time path-planning algorithm based on receding horizon techniques. *J. Intell. Robot. Syst.* 91 (3), 445–457.
- Naeem, W., Irwin, G.W., Yang, A., 2012. Colregs-based collision avoidance strategies for unmanned surface vehicles. *Mechatronics* 22 (6), 669–678.
- Seder, M., Petrovic, I., 2007. Dynamic window based approach to mobile robot motion control in the presence of moving obstacles. In: *Proceedings 2007 IEEE International Conference on Robotics and Automation*. IEEE, pp. 1986–1991.
- Singh, Y., Sharma, S., Sutton, R., Hatton, D., Khan, A., 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 Eng.* 169, 187–201.
- Tsou, M.-C., 2016. Multi-target collision avoidance route planning under an ecdis framework. *Ocean Eng.* 121, 268–278.
- Wang, X., Feng, K., Wang, G., Wang, Q., 2021. Local path optimization method for unmanned ship based on particle swarm acceleration calculation and dynamic optimal control. *Appl. Ocean Res.* 110, 102588.
- Yang, S., Li, L., Suo, Y., Chen, G., 2007. Study on construction of simulation platform for vessel automatic anti-collision and its test method. In: *2007 IEEE International Conference on Automation and Logistics*. IEEE, pp. 2414–2419.
- Zeng, W., Church, R.L., 2009. Finding shortest paths on real road networks: the case for A. *Int. J. Geogr. Inf. Sci.* 23 (4), 531–543.
- Zheng, H., Negenborn, R.R., Lodewijks, G., 2017. Fast ADMM for distributed model predictive control of cooperative waterborne AGVs. *IEEE Trans. Control Syst. Technol.* 25 (4), 1406–1413.
- Zhou, Y., Daamen, W., Vellinga, T., Hoogendoorn, S., 2019. Review of maritime traffic models from vessel behavior modeling perspective. *Transp. Res. C* 105, 323–345.
- Zhu, X., Yan, B., Yue, Y., 2021. Path planning and collision avoidance in unknown environments for USVs based on an improved D\* lite. *Appl. Sci.* 11 (17), 7863.