EURASIP Journal on Wireless Communications and Networking

RESEARCH

Open Access

A second-order dynamic and static ship path planning model based on reinforcement learning and heuristic search algorithms

Junfeng Yuan[1,2], Jian Wan[1,3], Xin Zhang[1,2*], Yang Xu[1,2], Yan Zeng[1,2] and Yongjian Ren[1,2*]

*Correspondence: zhangxin@hdu.edu.cn; yongjian.ren@hdu.edu.cn

[1] School of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou 310018, China
[2] Key Laboratory of Complex Systems Modeling and Simulation, Ministry of Education, Beijing, China
[3] School of Information and Electronic Engineering, Zhejiang University of Science and Technology, Hangzhou 310023, China

Abstract

Ship path planning plays an important role in the intelligent decision-making system which can provide important navigation information for ship and coordinate with other ships via wireless networks. However, existing methods still suffer from slow path planning and low security problems. In this paper, we propose a second-order ship path planning model, which consists of two main steps, i.e., first-order static global path planning and second-order dynamic local path planning. Specifically, we first create a raster map using ArcGIS. Second, the global path planning is performed on the raster map based on the Dyna-Sarsa(λ) model, which integrates the eligibility trace and the Dyna framework on the Sarsa algorithm. Particularly, the eligibility trace has a short-term memory for the trajectory, which can improve the convergence speed of the model. Meanwhile, the Dyna framework obtains simulation experience through simulation training, which can further improve the convergence speed of the model. Then, the improved ship trajectory prediction model based on stacked bidirectional gated recurrent unit is used to identify the risk of ship collision and switch the path planning from the first order to the second order. Finally, the second-order dynamic local path planning is presented based on the FCC-A* algorithm, where the cost function of the traditional path planning A* algorithm is rewritten using the fuzzy collision cost membership function (fuzzy collision cost, FCC) to reduce the collision risk of ships. The proposed model is evaluated on the Baltic Sea geographic information and ship trajectory datasets. The experimental results show that the eligibility trace and the Dyna learning framework in the proposed model can effectively improve the planning efficiency of the ship's global path planning, and the collision risk membership function can effectively reduce the number of collisions in A* local path planning and thus improve the navigation safety of encountering ships.

Keywords: Ship path planning, Ship navigation, Sailing safety, Heuristic search

1 Introduction

In recent years, economic globalization has been developing rapidly. As the backbone of international trade and the global economy, maritime transport carries over 80% of the volume of the international trade in goods is carried by sea, and the percentage is even higher for most developing countries.[1] Therefore, shipping has become more and more important, particularly where prosperity depended primarily on international trade.

However, the rapid development of the international shipping at sea makes the traffic condition increasingly complicated, and the world's main shipping routes and ports have formed complex networks, which are more prone to marine traffic accidents [1], and it pose a major threat to the safety of life and property at sea. Relevant statistics show that about 80%~85% of marine accidents are caused by human factors. For example, the ship drivers did not operate in accordance with regulations [2]. Although the International Maritime Organization has formulated the international rules for preventing collisions at sea, providing navigation methods and rules for ships at sea [3], and minimizing collisions between ships, it is still difficult to effectively reduce the probability of collisions only by relying on the experience of the crew.

As the level of marine technology improved, ships have been developing to be quite large-scale, specialized, and intelligent. Meanwhile, researchers have paid more attention to the research and development of intelligent decision-making systems for ship navigation. Particularly, the automated and intelligent driving systems can provide navigation for ship and coordinate with other ships via wireless networks, which can effectively reduce the occurrence of marine accidents Therefore, in order to ensure the safety of ships, drivers and the marine environment, it is extremely important to study related technologies of ship navigation intelligent decision-making system, and ship navigation path planning is one of the core links of this intelligent decision-making system [4]. Particularly, the ship's navigation intelligent decision-making system can provide the crew with important maneuvering suggestions for complex situations, and the ship's path planning is an important prerequisite for the motion control of the intelligent decision-making system and the output information of the intelligent decision-making system. Still, existing research works on the ship's path planning still suffer from challenges in the following two aspects:

- Scenario modeling of ship path planning and collision domain modeling. In many existing ship path planning researches in discrete scenarios, the simulation environment used for training is a simulated environment where obstacles are randomly generated. Note that the simulated environment is quite different from the real sea environment, and it is difficult to reflect the performance of the path planning model in the actual sea environment. At the same time, in the related works about the local encounter of ships, the ships appear in the form of simple geometry such as points and circles in most cases, and the ship domain model that can reflect the collision distance is rarely used. Therefore, in such a scenario where the collision distance of the ship needs to be considered, existing path planning models suffer from poor collision avoidance effect.
- The efficiency and safety of ship path planning. The current related research works usually do not distinguish between the long-distance navigation with few obstacles and the emergency ship encounter. In other words, the same model is used to deal with these two different navigation scenarios, which limits their performance in ship

[1] https://unctad.org/webflyer/review-maritime-transport-2021.

path planning tasks. At the same time, existing path planning models based on traditional reinforcement learning algorithm has the problems of slow planning speed and frequent collisions in the process of ship path planning. Besides, the heuristic search-based models have difficulties in avoiding collisions when encountering ships dynamically and cannot ensure the safety of ships.

In this work, we focus on ship path planning of intelligent decision-making systems, and propose a second-order ship path planning model. Specifically, the proposed model consists of two main steps, i.e., first-order static global path planning and second-order dynamic local path planning. Firstly, we create a raster map using ArcGIS, and the global path planning is performed on the raster map based on the Dyna-Sarsa(λ) model, which integrates the eligibility trace and the Dyna framework on the Sarsa algorithm. Particularly, the eligibility trace has a short-term memory for the trajectory, which can improve the convergence speed of the model. Meanwhile, the Dyna framework obtains simulation experience through simulation training, which can further improve the convergence speed of the model. Then, the improved ship trajectory prediction model based on stacked bidirectional gated recurrent unit is used to identify the risk of ship collision and switch the path planning from the first order to the second order. Finally, the second-order path planning is implemented based on the FCC-A* algorithm, where the cost function of the traditional path planning A* algorithm is rewritten using the fuzzy collision cost membership function (fuzzy collision cost, FCC) to reduce the collision risk of ships. The proposed model is evaluated on the Baltic Sea geographic information and ship trajectory datasets. Extensive experiments are conducted, and the results show that the proposed model can effectively improve the planning efficiency of the ship's global path planning, and the number of collisions is effectively reduced.

The main contributions are summarized as follows:

- We propose a second-order ship path planning model based on Dyna-Sarsa(λ) for addressing the problem of slow marine scene modeling and the low safety of ship path planning.
- An ship trajectory prediction model based on Stacked-BiGRUs and a Fuzzy Collision Cost A* (FCC-A*)-based dynamic local path planning algorithm are designed and integrated together to identify the risk of ship collision and improve the safety of path planning model in the case of ship encounters.
- The proposed model is evaluated on the Baltic Sea geographic information and ship trajectory datasets, and the results demonstrate the proposed model's effectiveness in terms of path planning efficiency and navigation safety.

The rest of this paper is organized as follows. The related works are reviewed in Sect. 2. The proposed method is introduced in Sect. 3, and extensive experimental results and discussion are given in Sect. 4. Section 5 concludes the main contributions and future works.

Table 1 Summary and comparison of three types of path planning methods

Method	Representative model	Advantages	Disadvantages
Traditional method	Artificial potential field	The planned path is smooth and safe	May fall into a local potential field, i.e., a local optimal solution
Machine learning	Q-learning	It is a model-free method with strong adaptability and can deal with uncertain environments	Blind exploration prolongs training time, causing the agent to converge to the wrong solution
Heuristic search	A* algorithm	High search efficiency, good stability, and can quickly respond to scene changes	Ignoring the node constraints of the moving volume, and the path planning is relatively rough

2 Related work

The path planning technologies for agent [5] are widely used in the field of automation, including autonomous collision avoidance of service robots, formation of drones, autonomous vehicle navigation, and so on. All path planning methods can solve most of the planning problems of point-line networks. According to the classification in common fields, path planning algorithms can be roughly divided into three categories: traditional path planning algorithms, machine learning algorithms, and heuristic search algorithms. Specifically, the advantages and disadvantages of three kinds of path planning algorithms are listed in Table 1.

2.1 Traditional path planning algorithms

As one traditional path planning method, simulated annealing (SA) [6] can solve the problem of finding the optimal solution in a limited range. Xu et al. [7] investigate the transportation efficiency and sales cost of the aquatic product market in Haikou of China, and use the SA algorithm to improve the aquatic product transportation route planning model, so that the model can find a low-cost transportation route. Xiao et al. [8] propose a coverage path planning method for UAVs to achieve full coverage of a target area and to collect high-resolution images while considering the overlap ratio of the collected images and energy consumption of clustered UAVs. However, the SA algorithm, which is a popular evolutionary algorithm widely used in dynamic path planning, suffers from high computation complexity problem. Therefore, Miao et al. [9] develop an enhanced SA approach by combining two additional mathematical operators and initial path selection heuristics into the standard SA. Particularly, the proposed model can perform robot path planning in dynamic environments with both static and dynamic obstacles, the computing performance of the standard SA is significantly improved while the generated solution is optimal or near-optimal. The improvement makes the proposed model being able to be applied in many real-time and online applications. Bedsides, Wang et al. [10] model the certain climbing ability and crossing ditch capability of the ground robot. Specifically, the authors proposed a model to search the shortest path from the start point to the end point, with reliable obstacle avoidance in the three-dimensional environments. Particularly, ant colony algorithm and genetic algorithm are integrated into the proposed model for improving the performance.

Moreover, the artificial potential field (APF) [11] method can construct virtual gravitational and repulsive forces. Specifically, the force between the end point and the object is the gravitational force, and the force between the object and the obstacle is the repulsive force. Therefore, we can set the force as a function for path optimization. Zhu et al. [12] propose a novel collision avoidance (CA) model by devising the APF method, and the proposed model is used to implement a practical ship automatic CA system. Particularly, in the proposed multi-ship CA model, the repulsive force model of APF is devised to incorporate the International Regulations for Preventing Collisions at Sea and the motion characteristics of the ship. Besides, inspired by navigation practice, the distance between the closest point of approach time and approach criterion is used as the unique changeable parameter.

Feng et al. [13] propose a new collision avoidance algorithm consisting of two main components, i.e., the path planning and the tracking controller. Specifically, a lateral lane-changing spacing model and the longitudinal braking distance model are designed to model the real vehicle's dynamic scenarios. Next, the authors incorporate the safety distance in a simulated traffic scene into the APF algorithm. Besides, the repulsion in the proposed model includes the force of the position repulsion and the speed repulsion, which are divided according to the threat level. At last, a predictive control model is designed to track the lateral motion through steering angle. Besides, the author present a Fuzzy-PID control to track the longitudinal speed, and the planned path is converted into an actual trajectory with stable vehicle dynamics. Vagale et al. [14] review guidance, and more specifically, path planning algorithms of autonomous surface vehicles and their classification, and provided potential need for new regulations for autonomous surface vehicles.

2.2 Machine learning algorithms

The idea of ant colony algorithm (ACA) [15, 16] draws on the foraging behavior of ants. Specifically, all ants smear their own pheromones on the roads they pass through in the process of searching for food. The road with food will be smeared with pheromone by multiple ants in a short time, so the concentration of pheromone will increase in a short time. The ants will choose the path according to the concentration of pheromone, and finally find the shortest path. In the online logistics scenario, the use of the responsive ant colony-based optimization algorithm has a good effect on the path planning problem of dense vehicles [17]. Particularly, the vehicle response speed can be improved by generating a diverse pheromone matrix. At the same time, the incorporation of simplified pheromone diffusion model, unequal distribution pheromone initialization strategy, and adaptive pheromone update mechanism into the ant colony algorithm can significantly enhance the computational speed and path quality of the classical ant colony algorithm [18].

Genetic algorithms (GAs) [19, 20] can simulate biological evolution, and is also an iterative search algorithm based on the principle of genetic genetics. Pehlivanoglu et al. [21] propose initial population enhancement methods in GA, and thus accelerate convergence process in the path planning problem of autonomous UAV. Nadia et al. [22] use a modified selection operator instead of using mutation operators, an adaptive population size and a modified procedure to perform a genetic algorithm, which outperformed other models in terms of distance minimization.GA is also widely used in multi-vessel collision avoidance scenarios [23, 24]. Particularly, GA-based model can meet the requirements of "early," "large," "wide" and "clear" for multi-vessel collision avoidance by incorporating ship navigation rules into genetic algorithms.

Reinforcement learning (RL) [25, 26] algorithm is a machine learning method in which the experimental target learns in the surrounding environment in a constantly trying way, and selects the next action according to the reward obtained by interacting with the environment. Therefore, the experimental target can obtain the maximum reward. In traditional path planning problems, reinforcement learning-based models use reward and punishment strategies to obtain optimal routes by continuously interacting with obstacles and passable areas. As for the problem of ship collision avoidance, Shen et al. [27] use the Bumper model in ship domain to incorporate avoidance experience into deep Q-learning based on maritime traffic rules, and rewrite the reward function part of the reinforcement learning algorithm. Therefore, the final collision avoidance model is in line with the actual ship motion and achieves good results in real ship collision avoidance experiments. Li et al. [28] investigate the path planning problem of USVs in uncertain environments, and proposed a path planning strategy unified with a collision avoidance function based on deep reinforcement learning (DRL).

Autonomous mobile robots usually move in dynamic unknown scenes, and can only plan paths through local information obtained from feedback, and their control quantities are continuous quantities. The gradient strategy algorithm A3C in reinforcement learning can handle the navigation problem in the continuous action space. However, the training time of A3C is quite long. Gao et al. [29] propose a new deep reinforcement learning (DRL)-based path planning model with incremental training for robot. Particularly, in order to deal with the complexity of real world applications, the authors combine twin-delayed deep deterministic policy gradients are with the traditional global path planning algorithm Probabilistic Roadmap to enhance the generalization ability of the proposed methods.

2.3 Heuristic search algorithms

The A* algorithm [30, 31] is widely used in various autonomous mobile robots and intelligent car navigation systems. Specifically, the algorithm can calculate the cost of each expansion node around it by selecting the corresponding heuristic function. Then, the position with the lower cost is selected as the next step by comparing the cost, until the target node position is found. Unmanned Surface Vehicles (USV) are widely used in modern cruises on the surface of water. In the study of intelligent navigation systems for unmanned vehicles, Song et al. [32] propose an improved A* algorithm that combines three path smoothing components, which reduces the path aliasing caused by the traditional A* algorithm. Experimental results show that the proposed algorithm achieves better performance than the traditional algorithm in both sparse and cluttered environments with uniform rasterization. The algorithm has been applied to the Springer USV navigation system. Guo et al. [33] propose a complete coverage path planning algorithm based on the improved A* algorithm to improve the efficiency and energy consumption of unmanned ships traversing the entire area. Singh et al. [34] present an A* approach for USV path planning in a maritime environment. Besides, the proposed approach is

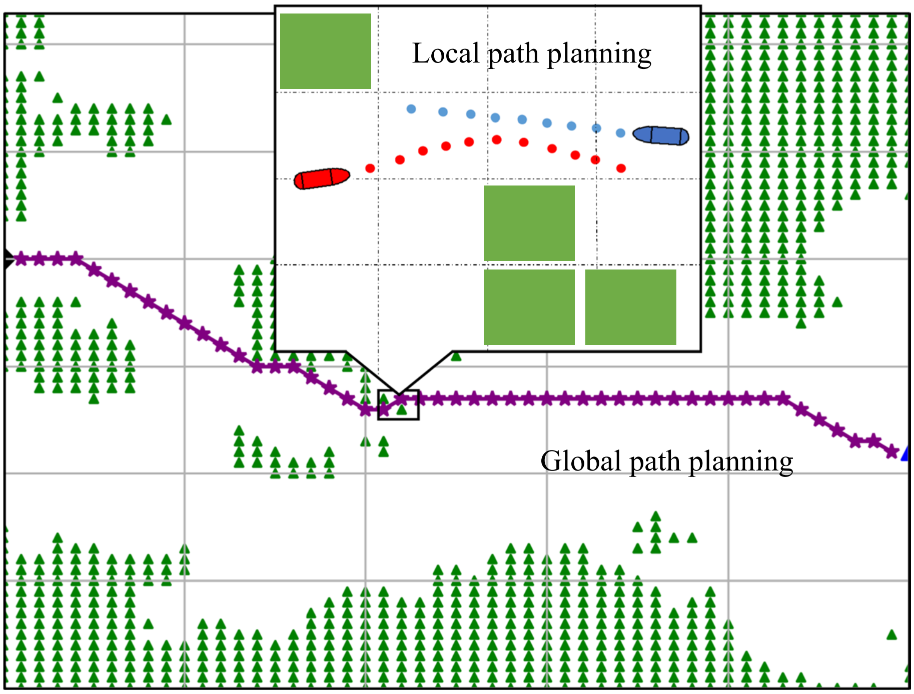


Fig. 1 Schematic diagram of second-order ship path planning. In the local path planning frame, the red ship should try to avoid collision with the blue ship that is sailing straightly. This planning method prioritizes the safety of the planned path and needs to avoid collisions with static obstacles and dynamic ships at the same time

extended to deal with the complex environments that are cluttered with static and moving obstacles and different current intensities.

3 Method

In this section, we introduce the proposed second-order ship path planning model in detail. Specifically, the problems that need to be solved in ship path planning are introduced first. Second, we present the modeling method of sea area scene, including the rasterization method and the storage format of geographic information. Then, the static global path planning algorithm based on Dyna-Sarsa(λ) is introduced, including the eligibility trace and the optimization process of Sarsa algorithm by Dyna framework. Finally, the dynamic local path planning algorithm based on Fuzzy Collision Cost A* (FCC-A*) is introduced, including the identification of collision risk, the construction of ship domain and the optimization process of collision risk membership function to A* algorithm.

3.1 Problem description

The proposed second-order ship path planning model needs to solve two problems, i.e., static global path planning and dynamic local path planning. Figure 1 shows a schematic diagram of the proposed path planning model.

On the macro level, the ship is in a long-distance sea area with few obstacles. As shown by the purple trajectory in Fig. 1, the ship will navigate in a global path planning manner,

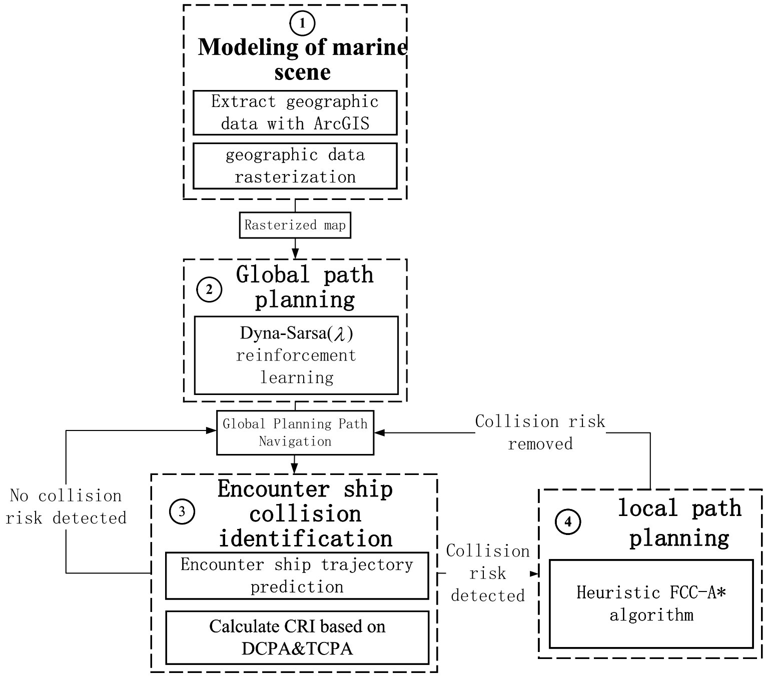

1
Modeling of marine scene
Extract geographic data with ArcGIS
geographic data rasterization
Rasterized map
2
Global path planning
Dyna-Sarsa(λ) reinforcement learning
Global Planning Path Navigation
Collision risk removed
No collision risk detected
3
Encounter ship collision identification
Encounter ship trajectory prediction
Calculate CRI based on DCPA&TCPA
Collision risk detected
4
local path planning
Heuristic FCC-A* algorithm

Fig. 2 Framework of second-order ship path planning model

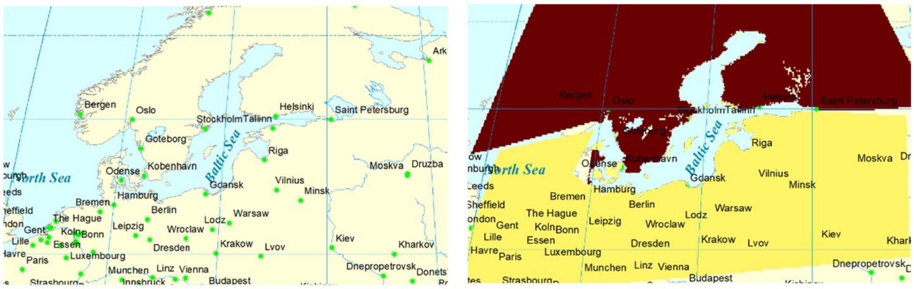
when there is no local dynamic ship collision risk. This proposed planning method gives more priority to the path planning speed and path length.

Microscopically, the ship trajectory prediction method based on Stacked-BiGRUs [35, 36] continuously detects the collision risk between the ship and other ships. The model will switch states and navigate in a local path planning manner when the ship collision risk index exceeds the rated threshold. As shown in Fig. 1, in the local path planning frame, the red ship should try to avoid collision with the blue ship that is sailing straightly. This planning method prioritizes the safety of the planned path and needs to avoid collisions with static obstacles and dynamic ships at the same time.

The framework of the proposed second-order ship path planning model is shown in Fig. 2, and it mainly consists of two components, i.e., global path planning and local path planning.

Global path planning. First, the ship path planning based on Sarsa reinforcement learning algorithm can effectively carry out path planning but the convergence speed is slow. Then, the eligibility trace and decay value mechanism are incorporated, and a global path planning algorithm based on the Sarsa(λ) [37] learning model is proposed. Finally, the reinforcement learning algorithm framework Dyna is presented. In particular, the global path planning speed is further accelerated by combining the Dyna framework and the Sarsa(λ) learning model into a Dyna-Sarsa(λ) learning model.

Local path planning. First, the ship collision risk identification is introduced. Specifically, the future route of the ship agent may inevitably collide with other dynamic ships when the ship is sailing on the globally planned path. At this time, the system should identify the collision risk and carry out dynamic local path planning to further ensure the navigation safety of the ship. This work focuses on the encounter situation of two



(a) Visualization Map (b) Raster Map

Fig. 3 Visualization map and raster map of the Baltic Sea geographic information.The rasterization method can represent geographic entities more effectively than the vector data method

ships, and the proposed model uses the ship trajectory prediction model to predict the future trajectories of the two ships in a period of time, and calculates the collision risk index (CRI) for each moment in this period of time. If the CRI index exceeds the threshold, the ship's path planning is switched to second order from the first order. Then, the traditional A* algorithm and its shortcomings that ignores the ship collision domain when applied to the ship trajectory planning problem is discussed. Finally, we introduce the GOODWIN ship domain model. The heuristic estimation cost of the A* algorithm is modified via the membership function, and the collision risk of dynamic obstacles is combined with the A* algorithm, and the FCC-A* path planning model is proposed to effectively reducing the collision risk of ships in local path planning.

3.2 Marine scene modeling

There are various methods for modeling geographic information of the marine scene, most of which are related to converting the surrounding environment into the problem of graph theory. The environmental map conversion methods in two-dimensional marine can be divided into vector data method and rasterization method, and their characteristics can be summarized as follows.

- Vector data have the advantages of standardized structure and low redundancy. Particularly, the data retrieval speed is fast, and the image resolution is high. However, the data structure is relatively complex, and it is difficult to process irregular graphics.
- Raster data have simpler data structure than that of vector data, and it is less difficult in spatial analysis or surface simulation. Besides, the integration or splicing of irregular graphics is more convenient, and it is easy to carry out various spatial analysis and mathematical simulation. The disadvantage is that the geographic information conversion becomes more difficult with data scale increasing.
- For the geographic features in large-scale ocean scenes shown in Fig. 3a, the rasterization method can represent geographic entities more effectively than the vector data method. The accuracy is determined by the grid side length.

In the sea area with complex weather and geographical environment, the ships may encounter many obstacles during the entire navigation process, and the obstacles include man-made marine structures, glaciers, reefs, etc. Such topographic data are generally stored in electronic charts. Therefore, it is necessary to convert the electronic chart into a scene data model that the algorithm can recognize to realize the path planning on the simulated electronic chart. The data source used in this paper is the shapefile format data based on ArcGIS, and the raster method is used to establish a static scene model with the vector data rasterization tool provided in ArcMap. This rasterization method belongs to an interpolation method, which is specially used to create a digital elevation model (DEM) that conforms to the real surface. The main principle of interpolation is to restore the real terrain by using traditional input data structures and known surface features.

Water is the primary erosive force that determines the general shape of most terrains. Therefore, most terrains contain many local maxima such as peaks, but few local minima, resulting in a discontinuous terrain state. Terrain to raster can constrain the interpolation process with surface-related constraints, generating a continuous terrain structure and an accurate representation of mountains and rivers. This type of function-constrained method can generate more accurate topographic maps with less input data. The scale of information will be smaller than the information required to describe geographic information with digital contours, further reducing the cost of obtaining accurate DEM.

This rasterization method is fully computed when removing sinks, and does not impose functional constraints where it might conflict with the input elevation data. Such conflicts are usually saved in log files in the form of sinks. These data can be used to correct geographic information, which is especially suitable for processing large and informative datasets such as marine environment. The rasterized Baltic Sea area is shown in Fig. 3b. Finally, the result data are saved in Shapefile format.

In particular, in the dynamic local ship path planning task, the extracted Shapefile data are used to model the local navigation chart, where the collision avoidance rules, navigation experience, ship operation characteristics and the size of the navigation chart should be fully considered. Let (x_s, y_s) be the starting position of the ship, and (x_d, y_d) be the target position at the end of the planning. Then, the center coordinate of the navigation chart point_{ce} is formally set as:

$$\text{point}_{\text{ce}} = \left(\frac{x_s + x_d}{2}, \frac{y_s + y_d}{2} \right). \tag{1}$$

The warp length l_{lon} and latitude length l_{lat} of the navigation chart are set as:

$$l_{\text{lon}} = |y_s - y_d|, \tag{2}$$

$$l_{\text{lat}} = |x_s - x_d|. \tag{3}$$

The grain size of rasterization determines the fineness of path planning, and it is necessary to coordinate the execution time of the algorithm and the planning quality.

3.3 Static global path planning algorithm based on Dyna-Sarsa(λ)

In this section, we will introduce the static global path planning algorithm based on Dyna-Sarsa(λ) in the proposed model. The main feature of the Sarsa algorithm is to perform single-step update. The value function is updated immediately after each step in the environment, which can quickly respond to environmental information. Therefore, the traditional Sarsa algorithm is represented as Sarsa(0). However, in the single-step update method, only the previous step that reaches the goal is related to the goal and all actions before that become unrelated. In particular, this situation will slow down the convergence speed of the algorithm. Generally, the continuous multi-step can be set as one round by extending the number of steps to update, and a complete update is conducted at the end of each round. This memory state of the continuous multi-step is called the eligibility traces (ET).

ET is an important concept in reinforcement learning. Sutton and Barto [38] pointed out that ET is additional memory variables associated with each state considering the frequency of visiting each state. There are three different expressions of ET: accumulating trace (AT), replacing trace (RT), and true online trace (TOT). The cumulative eligibility traces of state-action pairs are calculated as follows:

$$e_{t+1}(s,a) = \begin{cases} \gamma\lambda e_t(s,a) & \text{if } s \neq s_t, a \neq a_t \\ \gamma\lambda e_t(s,a) + 1 & \text{if } s = s_t, a = a_t \end{cases}, \tag{4}$$

where γ represents the discount factor. $\lambda \in [0, 1]$ is the decay coefficient of the trace, which defines how much the information of a selection in the past should be attenuated. In many cases, related studies have found that eligibility traces can speed up the convergence rate [39]. We can get the obtain the Sarsa(λ) algorithm by using the eligibility trace to modify the Sarsa algorithm, and Sarsa(λ) is shown in Algorithm 1.

Algorithm 1 Sarsa(λ) Algorithm

```
Input: discount factor $\gamma$, learning factor $\alpha$, decay coefficient $\lambda$, the count of
    iterations $\varepsilon$
Output: optimal strategy
Initialization: $Q(s,a)$, for all $s \in S, a \in A(s)$;
Repeat (for each episode in $\varepsilon$)
    $e(s,a) = 0$ for all $s \in S, a \in A(s)$; // initialize eligibility trace $e$
    Initialize S,A;
    Repeat (for each $t$ of episode) ; // if current state $s$ is not final state
        Take action $a_t$, observe $r(s_t, a_t), s_{t+1}$;
        Choose $a_{t+1}$ from $s_{t+1}$ using policy derived from Q(e.g., $\epsilon \cdot greedy$);
        $\delta \leftarrow r(s_t, a_t) + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)$;
        $e(s_t, a_t) \leftarrow e(s_t, a_t) + 1$;
        $Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha\delta e(s_t, a_t)$;
        $e(s_t, a_t) \leftarrow \gamma\lambda e(s_t, a_t)$;
        $s_t \leftarrow s_{t+1}$; $a_t \leftarrow a_{t+1}$; // update the current state and action
Until $s$ is terminal; // reach final state, Q table converges
```

In Algorithm 1, δ represents the temporal difference learning error (TD-error). At each moment, the current δ is assigned to each state according to its eligibility trace.

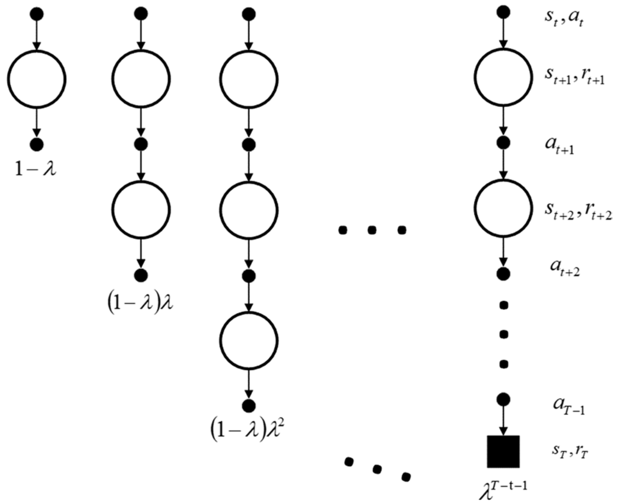


Fig. 4 Action trajectory diagram of Sarsa(λ). The state-action trajectory diagram of the Sarsa(λ) algorithm

The use of the eligibility trace allows the Sarsa(λ) algorithm to converge to the global optimum faster than the traditional Sarsa(0) algorithm. However, this acceleration is based on the preservation of past visits, and it will consume additional memory space. In the case with sufficient computing resources, choosing the Sarsa(λ) algorithm can quickly obtain a safer navigation planning path. The state-action trajectory diagram of the algorithm is shown in Fig. 4, where T is the total number of iterations.

The use of cumulative eligibility trace and decay coefficient λ in the optimization of the Sarsa algorithm can improve the convergence speed of the algorithm. However, the Sarsa(λ) algorithm still belongs to the category of model-independent reinforcement learning algorithms. In particular, the ships directly use the experience learned from the marine environment to generate, and the learning efficiency of this method is relatively limited. In model-based reinforcement learning algorithms, ships use the experience generated in the simulated environment to select new strategies by continuously refining the model.

During the training process with the Dyna learning framework, the ship first interacts directly in the simulation environment to obtain real experience to generate a pre-model, and interactively obtains simulation experience in the simulation scene inside the model at the same time. Besides, the real experience and simulation experience are integrated to train the ship, helping the ships plan and judge the optimal path. The core idea of the Dyna learning framework is to consume computing resources in exchange for high sampling efficiency. Particularly, more environmental interaction experience can be obtained, which improves the efficiency of the algorithm per unit time, while consuming computing resources. At the same time, in the stage of obtaining simulation experience in the Dyna model, the update method of the Q-learning algorithm is used. This method has the ability to learn the global optimum and can help the ship to avoid the local optimum situation. The steps of the Dyna-Sarsa(λ) algorithm after combining the Dyna learning framework with Sarsa(λ) are shown in Algorithm 2.

Algorithm 2 Sarsa(λ) Algorithm

```
Input: discount factor $\gamma$, learning factor $\alpha$, decay coefficient $\lambda$, Dyna simulation times $n$, iteration count $\varepsilon$
Output: optimal strategy $\pi$
Initialize $Q(s,a)$ and $Dyna\_Model(s,a)$; //initialize Q table and Dyna model
Repeat (for each episode in $\varepsilon$):
    $e(s,a) = 0$ for all $s \in S, a \in A(s)$; // initialize eligibility trace $e$
    Initialize S,A;
    Repeat (for each $t$ of episode): // if current state $s$ is not final state
        Take action $a_t$, observe $r(s_t,a_t), s_{t+1}$;
        Choose $a_{t+1}$ from $s_{t+1}$ using policy derived from Q(e.g., $\in \cdot greedy$);
        $\delta \leftarrow r(s_t,a_t)+\gamma Q(s_{t+1},a_{t+1})-Q(s_t,a_t)$; // Sarsa($\lambda$) learning process
        $e(s_t,a_t) \leftarrow e(s_t,a_t)+1$;
        $Q(s_t,a_t) \leftarrow Q(s_t,a_t)+\alpha\delta e(s_t,a_t)$;
        $e(s_t,a_t) \leftarrow \gamma\lambda e(s_t,a_t)$;
        $Dyna\_Model(s,a) \leftarrow r(s_t,a_t), s_{t+1}$;
        Repeat $n$ times:
            $s_t \leftarrow$random previously observed state;
            $a_t \leftarrow A(s_t)$; // randomly select action $a_t$ based on state $s_t$
            $r(s_t,a_t), s_{t+1} \leftarrow Dyna\_Model(s_t,a_t)$;
            $Q(s_t,a_t) = Q(s_t,a_t)+\alpha(r(s_t,a_t)+\gamma maxQ(s_{t+1},a_{t+1})-Q(s_t,a_t))$;
```

By integrating the Dyna framework with Sarsa(λ), the ships can not only obtain experience from the simulation training of the Dyna framework, but also learn experience from the direct interaction with the marine environment. The fusion of two kinds of experiences can provide guidance for ship path planning, which can greatly improve the efficiency of ship static global path planning.

3.4 Dynamic local path planning algorithm based on FCC-A*

It is necessary to switch from global path planning to local path planning for collision avoidance operations when a ship faces a collision risk. Therefore, this section first introduces the method for identifying the collision risk of encountering ships.

In the traditional autonomous ship collision avoidance system, the collision risk index (CRI) is usually used as an index to measure the collision risk of ships. The minimum value of CRI is 0 and the maximum value is 1. The minimum encounter distance (distance to closest point of approach, DCPA) and the minimum encounter time (time to closest point of approach, TCPA) are important factors for evaluating the CRI index between encountering ships in actual scenarios. As the value range of CRI has a nonlinear negative correlation with DCPA and TCPA, we use DCPA and TCPA to quantify CRI.

In consideration of the calculation of the collision risk of two ships, it is assumed that the status of the two ships at a certain moment is: $V_0\left(Lon_0, Lat_0, Sog_0, Cog_0\right)$ and $V_1\left(Lon_1, Lat_1, Sog_1, Cog_1\right)$, where *Lon*, *Lat*, *Sog*, and *Cog* represent the longitude, latitude, ground speed, and ground angle of the ship, respectively. Therefore, the relative speed S_r and relative angle C_r of the two ships at this moment can be calculated as:

$$S_r = \sqrt{Sog_0^2 + Sog_t^2 + 2Sog_0 Sog_t \cos\left(Cog_t - Cog_0\right)}, \tag{5}$$

$$cr = \begin{cases} Cog_0 - \text{arcos}\left(\frac{S_r^2 + Sog_0^2 - Sog_t^2}{2S_r Sog_0}\right) & Cog_0 < Cog_t \\ Cog_0 + \text{arcos}\left(\frac{S_r^2 + Sog_0^2 - Sog_t^2}{2S_r Sog_0}\right) & Cog_0 \geq Cog_t \end{cases}. \tag{6}$$

Besides, DCPA and TCPA are defined as:

$$\text{DCPA} = dist * \left(\sin\left(C_r - Cog_0 - \text{Bearing} - \pi\right)\right), \tag{7}$$

$$\text{TCPA} = dist * \left(\cos\left(C_r - Cog_0 - \text{Bearing} - \pi\right)/S_r\right), \tag{8}$$

where *dist* is the distance between the two ships on the sea, *Bearing* is the angle of the ship V_1 relative to V_0 when the ship V_0 is the coordinate origin. Besides, the unit of DCPA is nautical miles, and the unit of TCPA is minutes.

The relationships between CRI and DCPA or TCPA are defined as:

$$CRI_d = a_d \exp\left(b_d \text{DCPA}\right), \tag{9}$$

$$CRI_t = a_t \exp\left(b_t \text{TCPA}\right), \tag{10}$$

where the parameters a and b are the adjustment coefficients estimated according to the opinions of the ship experts and the watchmen in the ship transportation system. In this work, the parameters are set as $(a_d, b_d, a_t, b_t) = (1.0529, -1.5694, 1.3971, -0.0879)$ according to the movement of the objects on the sea [40]. CRI is calculated by the following formula based on the weighted sum of CRI_d and CRI_t

$$CRI = \alpha CRI_d + \beta CRI_t, \tag{11}$$

the parameters α and β are the weights of CRI_d and CRI_t, respectively. The sum of α and β is 1, and its value can be set according to the specific characteristics of marine traffic applications.

Whenever the state of the two ships at the next moment is predicted, the collision risk index CRI is calculated. If the CRI index exceeds the collision threshold, the ship changes from the global path planning state to the local path planning state.

In the local path planning stage, the collision model of the ship itself becomes a factor that cannot be ignored. The basic structure of the ship is shown in Fig. 5. Experts and scholars have conducted related research and proposed ship domain models suitable for different scenarios. In this work, we will first introduce the GOODWIN ship domain model.

Japanese ship expert FUJII first proposed the concept of ship domain in the 1960s. FUJII uses sensing equipment to collect and organize ship encounter behaviors in

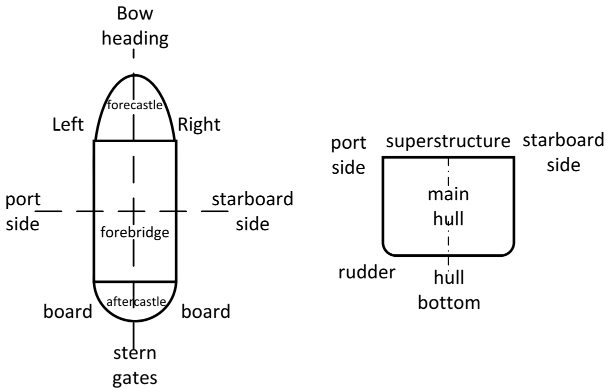


Fig. 5 Basic structure of the ship.

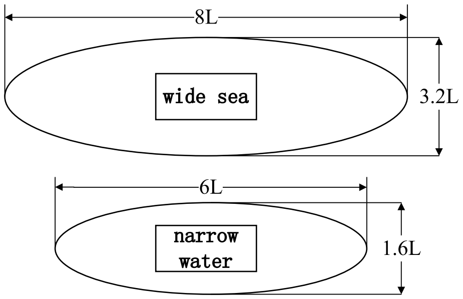


Fig. 6 FUJII ship domain model.The schematic diagram of the FUJII ship domain model proposed by Japanese ship expert FUJII in the 1960s

coastal waterways and crowded areas. Then, the ship collision avoidance trajectory data are filtered and analyzed, and finally an elliptical ship field is obtained. The ship is located at the intersection of the long and short axes. Specifically, the long axis is 8 times the length of the deck, and the short axis is 3.2 times the length of the deck. The schematic diagram of the FUJII ship domain model is shown in Fig. 6:

Then, GOODWIN improved the FUJII model into an asymmetrical shape via marine traffic surveys and a large number of collision avoidance experiments are conducted on radar simulators using crew training machines, taking into account the International Regulations for Preventing Collisions at Sea. The GOODWIN ship domain model with asymmetric shape based on the FUJII model. The model consists of three sectors with different radii spliced together. The sector areas are distributed according to the range of the ship's lights. Its fan-shaped radii are 0.7 nautical miles, 0.85 nautical miles and 0.45 nautical miles, respectively. The schematic diagram of the GOODWIN ship domain model is shown in Fig. 7.

The GOODWIN model is considered to be suitable for collision avoidance of ships at sea [40]. Particularly, the GOODWIN model is safer than the COLDWELL model and the FUJII model in practical use, so the GOODWIN model is selected as the collision domain model in this study. The GOODWIN model calculates the ship domain according to the angle relationship between ships. Formally, GOODWIN is defined as:

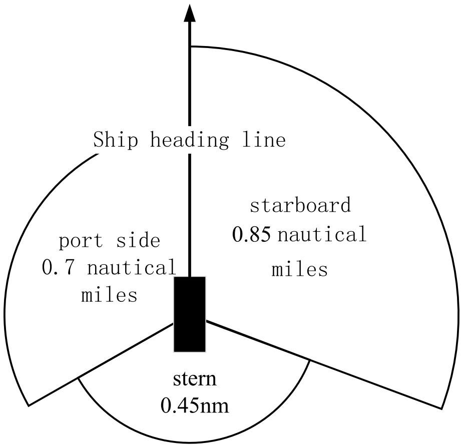


Fig. 7 GOODWIN ship domain model.The schematic diagram of the GOODWIN ship domain model, which is an improved version of FUJII ship domain model

$$\text{GOODWIN} = \begin{cases} 0.85 & 0^\circ \le \theta < 112.5^\circ \\ 0.45 & 112.5^\circ \le \theta \le 247.5^\circ \\ 0.7 & 247.5^\circ < \theta < 360^\circ \end{cases}. \tag{12}$$

In this work, the A* algorithm will be used to obtain the local optimal planning path of the ship. The cost function $f(k)$ of the A* algorithm in this scenario should be expressed as the sum of the navigation distance cost and the collision cost:

$$f(k) = g(k) + h(k), \tag{13}$$

where $g(k)$ is the cost of the ship's distance from the starting point, and its initial value is 0. Besides, the heuristic cost function $h(k)$ can choose from a variety of methods to calculate the distance, such as Manhattan distance, Euclidean distance, and Chebyshev distance. Considering the underactuated characteristics of the ship (the degree of freedom of the ship's navigation is less than the degree of freedom of the marine environment) [27], we use the sum of the Chebyshev distance and the collision cost (fuzzy collision cost, FCC) at this point as the heuristic estimated cost of (k). Give the ship a guiding direction, and the specific calculation expression of (k) is:

$$(k) = \max\left(|x_k - x_t|, |y_k - y_t|\right) + FCC(dist, \theta_1, \theta_2), \tag{14}$$

where (x_t, y_t) is the position coordinate of the target waypoint, (x_k, y_k) is the current position coordinate of the ship. Besides, we set the clockwise direction of true north from 0° to 360°, and the ship direction angle is the angle between the ship's bow and the true north direction. FCC is the fuzzy collision cost based on the GOODWIN ship domain model. Next, the collision cost FCC based on the fuzzy model is introduced.

The basic operation in traditional Boolean logic is "and, or, not," which is suitable for scenarios with clear logic. However, there is no particularly clear threshold when actually judging the distance and angle of two ships. In fuzzy logic, there are no strict boundaries between distances and angles, and the classification of different orientations is measured by the degree of membership. Specifically, the degree of membership refers to the quantitative analysis of a fuzzy research object through membership functions, and the process of transforming logical input values into membership degrees of each set is called fuzzification. The calculation of the collision risk membership function FCC can be expressed as:

$$\mathrm{FCC} = \frac{1}{2}U_{\theta} + \frac{1}{2}U_{\mathrm{dist}}, \tag{15}$$

where U_{θ} is the membership function of the azimuth angle θ between the current ship and the target ship, and U_{dist} is the membership function of the distance dist between the current ship and the target ship.

The collision risk index encountered by the ship will change with the relative angle of the two ships. U_{θ} is a function of the included angle between the two ships. According to the ship collision avoidance rule [41], the membership degree of the azimuth angle θ between the current ship and the target ship is defined as:

$$U_{\theta} = \frac{17}{44}\left[\cos\left(abs(\theta_1 - \theta_2) - 19^{\circ}\right) + \sqrt{\frac{440}{289} + \cos^2(abs(\theta_1 - \theta_2) - 19^{\circ})}\right]. \tag{16}$$

Moreover, *dist*, the distance between the ship and the target ship, will also cause the change of the collision risk index. Combined with the GOODWIN ship domain model, the surrounding of the ship is divided into three areas, and the collision risk U_dist is calculated for each area separately, which is shown in Algorithm 3.

Algorithm 3 Collision cost between two ships

```
Input: the distance between the two ships $dist$, the direction angle of the two ships $(\theta_1, \theta_2)$
Output: Collision cost value $U_{dist}$
Initialize delta=$|\theta_1 - \theta_2|$;
If $dist > 1.7$ miles:
    Return $U_{dist} = 0$; // the two ships are at a safe distance greater than twice the maximum sector radius
Elif $dist < 0.9$ miles:
    Return $U_{dist} = 1$; // the two ships are at a safe distance less than twice the minimum sector radius
Elif $67.5^{\circ} \leq delta < 180^{\circ}$:
    Return $U_{dist} = \left(\frac{1.55 - dist}{1.55}\right)^2$; //the situation where the starboard and starboard sectors of the two ships meet
Elif $247.5^{\circ} \leq delta < 360^{\circ}$:
    Return $U_{dist} = \left(\frac{1.15 - dist}{1.15}\right)^2$; //the port sector of the own ship faces the stern sector of the other ship
Elif $0^{\circ} \leq delta < 67.5^{\circ}$:
    Return $U_{dist} = \left(\frac{1.3 - dist}{1.3}\right)^2$; //the starboard sector of this ship faces the stern sector of other ships
```

Besides, we use the collision risk membership function FCC to modify the heuristic estimation cost function of the traditional A* algorithm, and the process of the FCC-A* algorithm is shown in Algorithm 4.

```
Algorithm 4 FCC-A* Algorithm
Input: starting point $n_s$, target point $n_g$, magnification factor $\alpha$, predicted
    trajectory sequence of direct ship $L$
Output: the path from $n_s$ to $n_g$
Function Cost$(n, n')$: // calculate the heuristic cost modified by the
    collision risk membership function
    If $n'$ is $n_s$:
        $f(n) = g(n) + max(|n.x - n_g.x|, |n.y - n_g.y|)$;
    Else:
        $\theta_1$ is the angle due north of the line between $n$ and $n'$;
        $\theta_2$ is the heading of stand-on vessel in $L$;
        $f(n) = g(n) + max(|n.x - n_g.x|, |n.y - n_g.y|) + FCC(dist, \theta_1, \theta_2)$;
    return $f(n)$
End function
Function updateState$(n, n')$: // the cost of updating the path
    $tmp(n, n') = |n.x - n'.x| + |n.y - n'.y|$;
    If $n'$ is an obstacle or $n' \in CLOSED$:
        ignore this $n'$;
    Elif $n' \in OPEN$:
       If $g(n') > g(n) + tmp(n, n')$:
            $parent(n') = n$;
           $g(n') = g(n) + tmp(n, n')$;
        Else:
            OPEN.Insert$(n', Cost(n, n'))$;
            $parent(n') = n$;
            $g(n') = g(n) + tmp(n, n')$;
End function
Function Main(): //FCC-A* algorithm starts executing
    Initialize $g(n_s) = 0$, $OPEN = \emptyset$, $CLOSED = \emptyset$;
    OPEN.Insert$(n_s, Cost(n_s, n_s))$:
        While $OPEN \neq \emptyset$ do:
            $n = OPEN.minCost(n)$;
            CLOSED=CLOSE$\cap n$;
            If $n = n_g$:
                return path;
            Else:
                $n'$ are the neighbors of $n$;
                for all $n'$ do:
                    updateState$(n, n')$;
        EndWhile
End function
```

4 Results and discussion

In this section, we conduct extensive experiments to evaluate the proposed ship path planning model in details. Specifically, we first analyze the global path planning performance of the proposed model based on Dyna-Sarsa(λ). Then, the local path planning based on FCC-A* is evaluated.

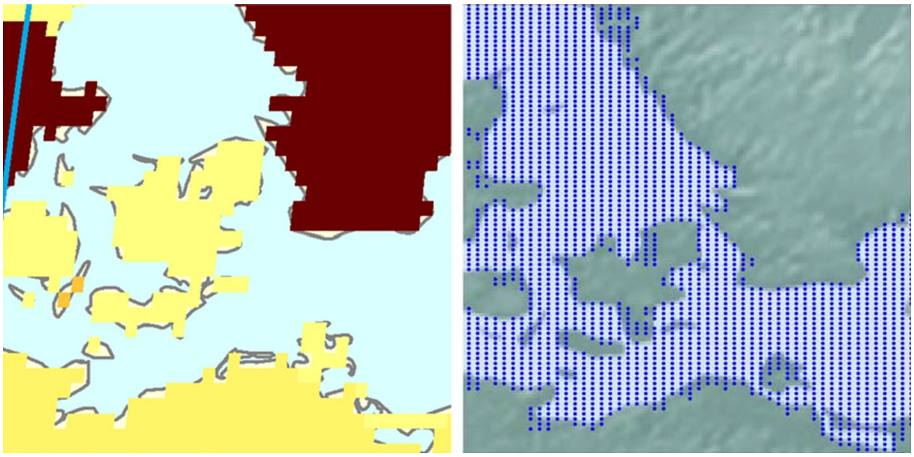


Fig. 8 Nautical chart of simulation experiments. This figure shows the simulation experimental chart by rasterizing the shapefile data model of part of the Baltic Sea, and we can observe that the experimental chart basically simulates the static obstacles in the sea area, which reflects the ability of the sea scene modeling

4.1 Analysis of global path planning based on Dyna-Sarsa(λ)

The simulation experimental chart by rasterizing the shapefile data model of part of the Baltic Sea is shown in Fig. 8. We can observe that the experimental chart basically simulates the static obstacles in the sea area, which reflects the proposed model's ability of the sea scene modeling.

The Q-learning algorithm, Sarsa algorithm, Sarsa(λ) algorithm and Dyna-Sarsa(λ) algorithm are introduced into the simulation chart for evaluation, and each algorithm was trained for 2000 rounds. In the main test of the Dyna-Sarsa(λ) learning algorithm, 40 rounds of simulations are performed using the Dyna learning framework, which means that the ship interacts with the simulated environment for 40 rounds to obtain simulation experiences. The experiment of each algorithm repeats 6 times, and the average value of the corresponding evaluation index is used as the final experimental result. The evaluation indicators of algorithm performance are the reward value of each iteration, the number of collisions per iteration, and the convergence speed of the algorithm. Then, we will evaluate the performance of the Dyna-Sarsa(λ) model. The baseline models are listed as follows:

1. Q-learning learning model. The Q-learning model is used as the benchmark reference model for the ablation experiments in this section.
2. Sarsa learning model. The Sarsa learning model is an online improvement in the Q-learning model, and it is more cautious in exploration than Q-learning.
3. Sarsa(λ) learning model. Sarsa(λ) is a learning model obtained by improving the round update method of Sarsa with eligibility traces.
4. Dyna-Sarsa(λ) learning model. The proposed model that uses the Dyna learning framework to enable the Sarsa(λ) learning model to gain simulation experience.

Figures 9 and 10 show the performance of different models on the simulation experiment charts, and the visualization results of path planning are shown in Fig. 11.

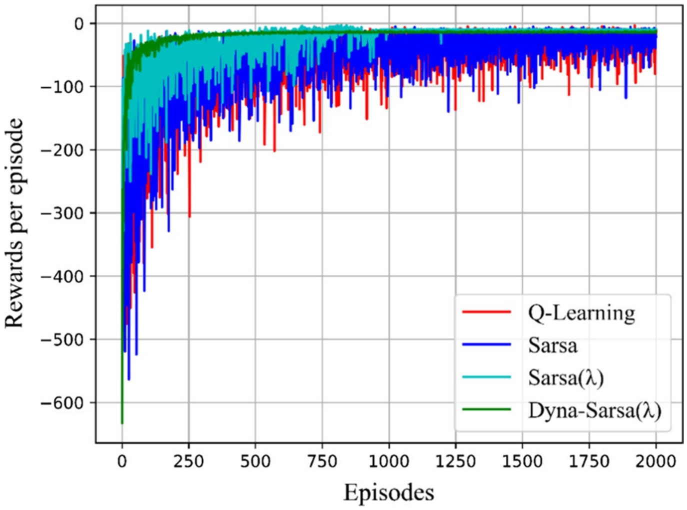


Fig. 9 Comparison of ship reward value for each iteration. The performance comparison of ship reward value achieved by the proposed model and baselines for each iteration

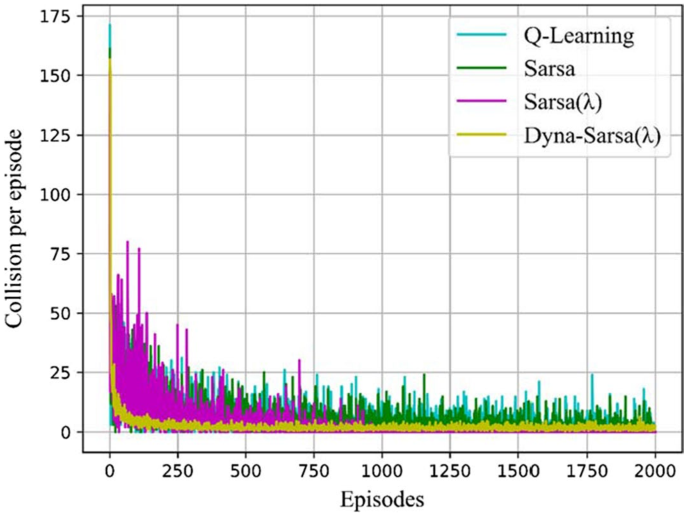


Fig. 10 Comparison of the number of ship collisions in each iteration. The performance comparison of the number of ship collisions between the proposed model and baselines in each iteration

Specifically, we can observe that: (1) the convergence speed and average reward of the Sarsa learning model and the Q-learning model are quite similar. However, the number of collisions per round of the Sarsa model is less than that of the Q-learning model with an average decrease of 9.8%, which indicates safer navigation. The main reason is that Sarsa is sensitive to the penalty value brought by the collision and adopts a more cautious strategy. Therefore, the safety of the Sarsa-based model is higher than that of the Q-learning model in the application of ship path planning. (2) The eligibility trace can effectively improve the convergence speed of the Sarsa model. The Sarsa(λ) model converges in about 900 rounds, and the corresponding Sarsa model and Q-learning model basically reach the convergence after 2000 rounds. The reason is that the eligibility trace can mark the value of the positions

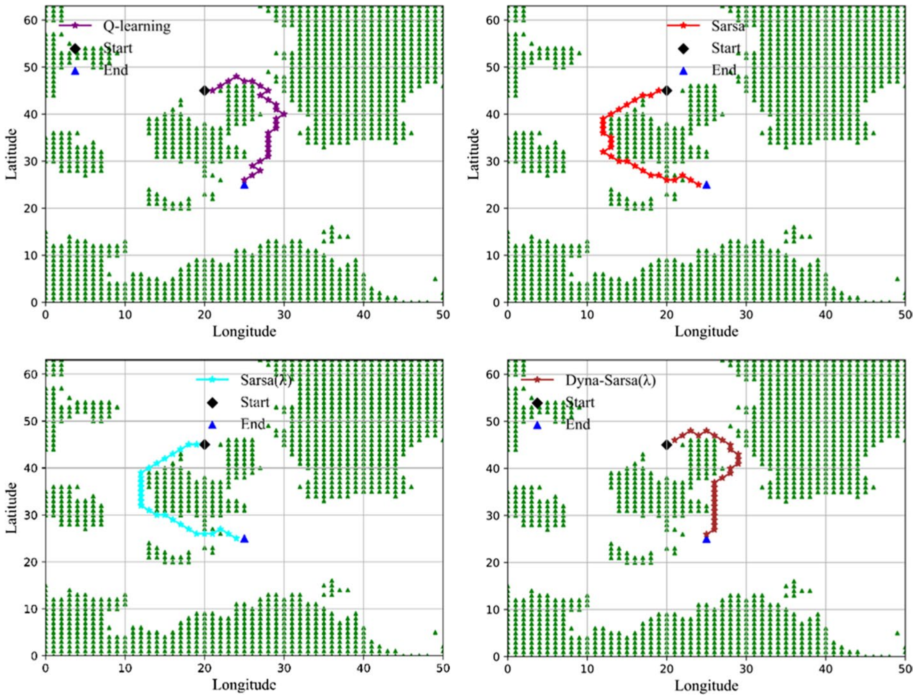


Fig. 11 Path planning simulation of four types of learning model. The visualization results of path planning simulation achieved by the proposed model and baselines

at different distances from the target point, which can guide the state transition selection in the subsequent rounds and help the ship to find the optimal solution faster. (3) The Dyna learning framework can improve the convergence speed of the Sarsa(λ) model. Specifically, compared with the Sarsa(λ) model, which converges in about 900 rounds, the Dyna-Sarsa(λ) learning model reaches the convergence state in about 500 rounds. Besides, the average number of collisions per round decreased by 73.3% compared with the Sarsa model. This is because the Dyna learning framework can help the Sarsa(λ) learning model to gain experience from the simulated environment, and the simulated experience can guide the ship to choose the optimal path. (4) As shown in Fig. 10, in the trajectory planning diagram of the four types of learning models, the Q-learning model tends to perform path planning through the right channel closer to the target point, while Sarsa and Sarsa(λ) are more likely to perform path planning through the wide left waterway. The results reflect that the Sarsa-related model is sensitive to collision risk and will abandon closer paths to avoid obstacles. The Dyna-Sarsa(λ) learning model uses the Q-learning algorithm in the stage of acquiring simulation experience, so it can plan shorter paths under the premise of ensuring safety.

In conclusion, the experimental results show that the learning model based on Sarsa has higher navigation safety, and both the eligibility trace and the Dyna learning framework can effectively improve the convergence speed in the experiments.

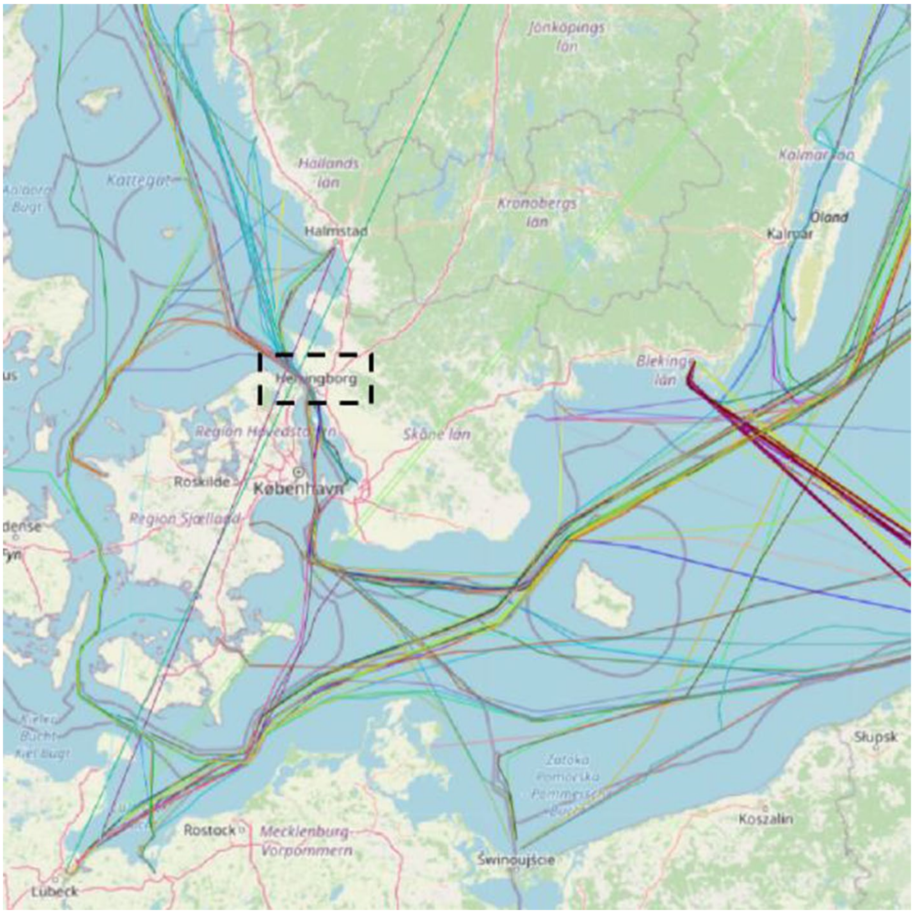


Fig. 12 Visualization of summer ship trajectories in the Baltic Sea. This figure shows the visualization of ship trajectories in part of the Baltic Sea. The dashed box is the port of Helsingborg, which has the characteristics of dense ships, complex historical trajectories, and high possibility of ship collision events

4.2 Analysis of local path planning based on FCC-A*

In order to provide a local path planning scheme when the ship encounters the danger of collision, we propose a dynamic local path planning model based on the FCC-A* algorithm with trajectory prediction for a collision risk identification method.

In this section, we first evaluate the method for calculating the collision risk of encountering ships. Then, the FCC-A* algorithm is evaluated with the path planning time and the number of path collisions. Specifically, we select the trajectory data of two encountering ships in the Baltic Sea summer ship trajectory data for evaluations.

Figure 12 shows the visualization of ship trajectories in part of the Baltic Sea. The dashed box is the port of Helsingborg, which has the characteristics of dense ships, complex historical trajectories, and high possibility of ship collision events. Therefore, we select the ship trajectory data that have performed the collision avoidance operation in this port as the experimental dataset.

Firstly, we train a Stacked-BiGRUs trained on the Baltic Sea summer ship trajectory dataset, and use the trained model to predict the trajectories of the two encountering ships. The prediction results are shown in Fig. 13. The blue trajectory in the figure is the historical trajectory of the blue ship, and its route direction is the direction of the blue

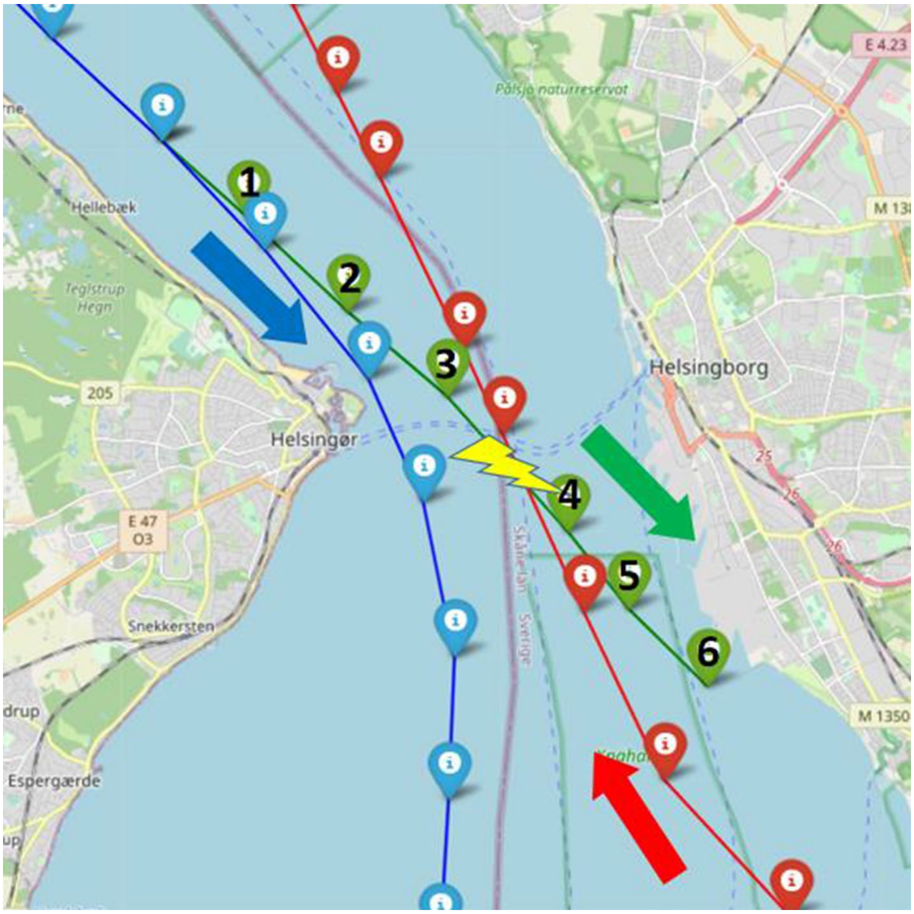


Fig. 13 Visualization of improved ship trajectory prediction (green trajectory) based on Stacked-BiGRUs. Firstly, we train a Stacked-BiGRUs trained on the Baltic Sea summer ship trajectory dataset and use the trained model to predict the trajectories of the two encountering ships. The prediction results of the two encountering ships by the pretrained model are shown in this figure. The blue trajectory in the figure is the historical trajectory of the blue ship, and its route direction is the direction of the blue arrow. Besides, the red line is the red ship's historical trajectory, and the sailing direction is the along the red arrow. The green trajectory is the predicted trajectory of the blue ship from a certain moment

arrow. Besides, the red line is the red ship's historical trajectory, and the sailing direction is the along the red arrow. The green trajectory is the predicted trajectory of the blue ship from a certain moment.

Since ship collision avoidance is an emergency event, the relevant trajectory data account for a very low proportion in the ship trajectory training set. Therefore, the trajectory predicted by the ship trajectory prediction model is the usual maneuvering behavior of the ship. Generally, the crew is not aware of the danger of collision during the actual navigation, and proceed according to the original route.

The red ships in Fig. 13 sail directly, and the blue ships should give way. If the blue ship does not perform the collision avoidance operation according to the blue trajectory in the figure, but continues to sail according to the predicted green trajectory, it will collide with the red ship at the yellow lightning mark and cause a marine traffic accident. In this situation, it is necessary to establish a collision risk identification mechanism first. Specifically, the collision risk index of the ship should be calculated

Table 2 Collision risk correlation index change

Index	Step1	Step2	Step3	Step4	Step5	Step6
DCPA	1.990	0.533	0.145	0.479	1.212	1.521
TCPA	17.39	10.21	6.89	13.84	16.33	19.92
CRI	0.174	0.512	0.801	0.455	0.244	0.169

in combination with the collision risk index by predicting the ship's future route. If the collision risk index of the ship exceeds the set threshold, it is determined that the ship has an accident risk, and collision avoidance operations need to be performed in this area.

Therefore, the ship collision risk should be firstly identified based on the improved ship trajectory prediction model. Since the closest encounter distance of the ship needs to be more than one nautical mile, the collision risk index (CRI) between the next six predicted positions of the direct ship and the give-way ship is calculated to identify collision risks in time and carry out local path planning. Table 2 shows the changes of DCPA, TCPA and CRI of the six predicted positions of the two ships. It can be seen that the CRI value of Step 3 corresponding to the yellow mark has reached about 0.8. Note that the value range of CRI is 0 to 1, and the larger the value, the higher the collision risk. Particularly, if the value exceeds 0.5, the ship has a collision risk [42]. Therefore, it can be seen from the table that at Step 2, the static global path of the ship needs to be switched to local dynamic path planning.

Before evaluating the FCC-A* algorithm, we first set the related parameters. The search range of the traditional A* algorithm is eight grids around a grid. However, the search direction is simplified, and only the three directions, i.e., front, left and right, are searched due to the under-driven property of marine ships. At the same time, the ship cannot turn in the direction that is a boundary or a marine obstacle. These constraints narrow the search scope of the algorithm, thereby reducing the algorithm execution time and ensuring that the planned path meets the navigation requirements of the ship.

As shown in Fig. 14, the collision avoidance scene is rasterized according to the sea area scene modeling method. The black squares in the figure represent the terrain of the sea area, which are static obstacles. Besides, the red squares are the ship sailing directly, and the blue squares are the ship that should make way. In this experiment, the direct-sailing ship simulates the navigation process by printing its historical trajectory in real time, and the goal of the avoidance ship is to reach the green star without colliding with the direct-sailing ship and obstacles.

The experimental results of the traditional A* algorithm are shown in Fig. 15, and the planning results of the FCC-A* algorithm are shown in Fig. 16. We can observe that the traditional A* algorithm can avoid all static obstacles well, and the path length is also optimal. However, it is difficult for traditional A* algorithm to effectively avoid the straight ships whose position changes dynamically in each round, resulting in the collision between the give-way ship and the direct-sailing ship at the yellow sign. Besides, the proposed FCC-A* algorithm considers the collision field between the give-way ship and the direct-sailing ship, and the collision risk is incorporated into the cost function of the A* algorithm as a membership function. Therefore, the

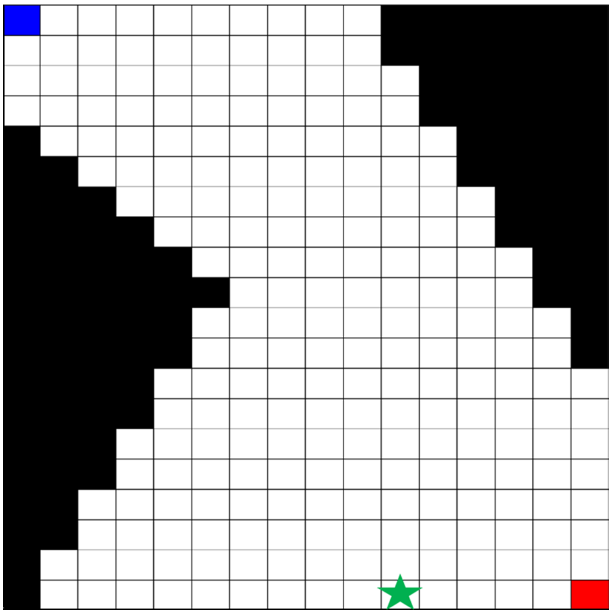


Fig. 14 Local dynamic path planning scenario. The collision avoidance scene is rasterized according to the sea area scene modeling method. The black squares in the figure represent the terrain of the sea area, which are static obstacles. Besides, the red squares are the ship sailing directly, and the blue squares are the ship that should make way

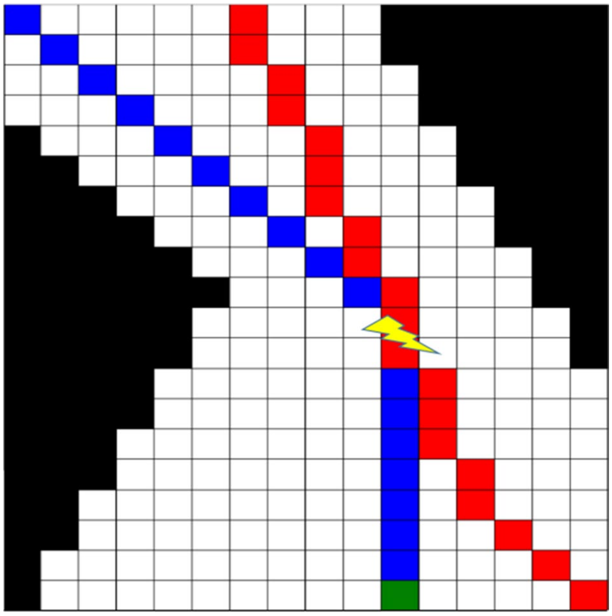


Fig. 15 Path planning results of traditional A* algorithm. The experimental results of the traditional A* algorithm are shown in this figure. We can observe that the traditional A* algorithm can avoid all static obstacles well, and the path length is optimal. However, it is difficult for traditional A* algorithm to effectively avoid the straight ships whose position changes dynamically in each round, resulting in the collision between the give-way ship and the direct-sailing ship at the yellow sign

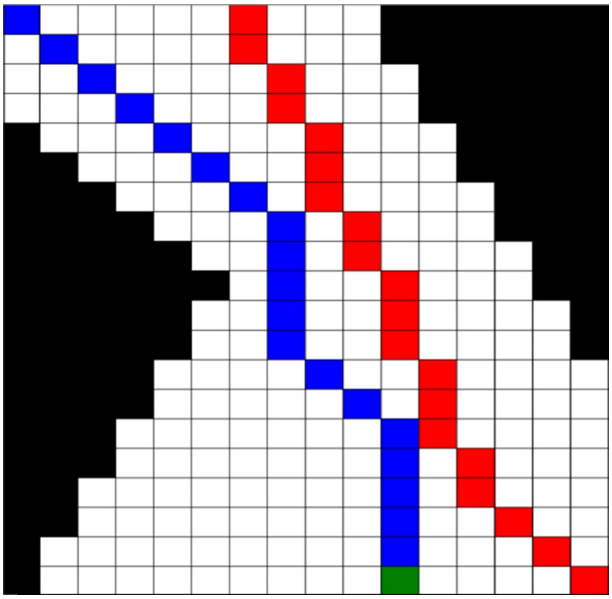


Fig. 16 Path planning results of FCC-A*. The planning results of the FCC-A* algorithm are shown in this figure. The FCC-A* algorithm considers the collision field between the give-way ship and the direct-sailing ship, and the collision risk is incorporated into the cost function of the A* algorithm as a membership function. Therefore, the give-way ship will consider the cost of collision with the direct-sailing ship in every step, thereby avoiding the risk of collision

Table 3 Comparison of planning time of two algorithms (unit: seconds)

Scene scale	100 grid	400 grid	900 grid	1600 grid	2500 grid	3600 grid
A*	0.104	0.160	0.235	0.446	0.879	1.023
FCC-A*	0.134	0.220	0.301	0.747	1.223	1.790

Table 4 Comparison of collision times between two algorithms (unit: times)

Scene scale	100 grid	400 grid	900 grid	1600 grid	2500 grid	3600 grid
A*	2	3	3	2	5	4
FCC-A*	0	0	0	0	0	0

give-way ship will consider the cost of collision with the direct-sailing ship in every step, and FCC-A* algorithm can help to avoid the risk of collision.

Furthermore, we compare the traditional A* algorithm and the FCC-A* algorithm in the case of two ships meeting on 6 local path planning experiments with different scales. The experimental results of the planning time are given in Table 3. It can be observed that the calculation time of the FCC-A* algorithm is about 30% higher than that of the traditional A* algorithm due to the component of the fuzzy model in FCC-A*. However, since the search speed of the A* algorithm is quite fast, the delay of the FCC-A* algorithm is acceptable.

Table 4 presents the comparison of the number of collisions between the two algorithms at different scales. In particular, the collision refers to the number of times the grid positions of the give-way ship and the direct-sailing ship overlap at the same time. We can observe that the path of the give-way ship planned by the traditional A* algorithm has 3 to 4 collisions with the direct ship on average, while the planned path of the FCC-A* algorithm basically has no collision. This is because the FCC-A* algorithm uses the membership function calculation to quantify the collision risk of the two ships and incorporate it as part of the cost calculation. Therefore, the dynamic path planning process considers the collision risk at each moment, which greatly reduces the collision risk between the give-way ship and the direct ship.

In conclusion, the experimental results show that the dynamic path local planning model based on FCC-A* has a slight loss in planning speed compared with the traditional algorithm, but its planned path is safer than that of the traditional algorithm.

5 Conclusion

In this work, a second-order ship path planning model is proposed to address the problem of sea area scene modeling and the slow speed and low safety of ship path planning. Specifically, we first create a raster map with ArcGIS, and the global path planning is performed on the raster map based on the Dyna-Sarsa(λ) model, which integrates the eligibility trace and the Dyna framework on the Sarsa algorithm. Particularly, the eligibility trace is adopted to improve the convergence speed of the model. Meanwhile, the Dyna framework obtains simulation experience through simulation training, which can further improve the convergence speed of the model. Then, the improved ship trajectory prediction model is used to identify the risk of ship collision and switch the path planning from the first order to the second order. Finally, the second-order dynamic local path planning is implemented based on the FCC-A* algorithm, where the cost function of the traditional path planning A* algorithm is rewritten using the fuzzy collision cost membership function to reduce the collision risk of ships. The proposed model is evaluated on the Baltic Sea geographic information and ship trajectory datasets, and the experimental results show the effectiveness of the proposed model.

In the future, we plan to adopt federated learning model [43] for privacy protection of each ship without influencing the path planning performance. Besides, collaborative learning of local and global features [44, 45] can be used to guide each ship to plan a safe path through the collaborative collision avoidance of multiple ships. Moreover, edge computing techniques [46] can be also applied in the field of ship path planning to further improve the perception ability and decision-making ability and improve the efficiency and safety of ship path planning.

Abbreviations

FCC	Fuzzy collision cost
BiGRU	Bidirectional gated recurrent unit
SA	Simulated annealing
APF	Artificial potential field
CA	Collision avoidance
ACA	Ant colony algorithm
GA	Genetic algorithm
RL	Reinforcement learning
DRL	Deep reinforcement learning
USV	Unmanned surface vehicle
CRI	Collision risk index
DEM	Digital elevation model
ET	Eligibility trace

AT	Accumulating trace
RT	Replacing trace
TOT	True online trace
TD-error	Temporal difference learning error
DCPA	Distance to closest point of approach
TCPA	Time to closest point of approach

Acknowledgements
Not applicable.

Author contributions
JY, JW, and YX proposed the path planning model and designed the system. XZ and YX implemented the simulation. JY, XZ, and YZ wrote the paper. JW, XZ, YZ, and YR revised the manuscript. All authors read and approved the final manuscript.

Funding
This study was supported in part by the Zhejiang Key Research and Development Program under grants 2021C03187, the Open Research Project Fund of Key Laboratory of Marine Ecosystem Dynamics, Ministry of Natural Resources under Grants MED202202, and the National Natural Science Foundation of China under Grants J2024009 and 62072146.

Availability of data and materials
The datasets used and/or analyzed during the current study are available from the corresponding author upon reasonable request.

Declarations

Competing interests
The authors declare that they have no competing interests.

Received: 25 July 2022 Accepted: 20 December 2022
Published online: 30 December 2022

References

1. X. Wu, L. Zhang, M. Luo, Current strategic planning for sustainability in international shipping. Environ. Dev. Sustain. **22**(3), 1729–1747 (2020)
2. C. Baker, D. McCafferty, Accident database review of human element concerns: what do the results mean for classification, in *Proceedings of International Conference on Human Factors in Ship Design and Operation*, RINA (2005). Citeseer
3. M.R. Benjamin, J.A. Curcio, Colregs-based navigation of autonomous marine vehicles, in *2004 IEEE/OES Autonomous Underwater Vehicles* (IEEE Cat. No. 04CH37578), pp. 32–39 (2004). IEEE
4. B. Wu, T. Cheng, T.L. Yip, Y. Wang, Fuzzy logic based dynamic decision-making system for intelligent navigation strategy within inland traffic separation schemes. Ocean Eng. **197**, 106909 (2020)
5. H.-y. Zhang, W.-m. Lin, A.-x. Chen, Path planning for the mobile robot: a review. Symmetry **10**(10), 450 (2018). https://doi.org/10.3390/sym10100450
6. R. Zeng, Y. Wang, A chaotic simulated annealing and particle swarm improved artificial immune algorithm for flexible job shop scheduling problem. EURASIP J. Wirel. Commun. Netw. **2018**(1), 1–10 (2018)
7. Q. Xu, J. Wang et al., Study on optimization of aquatic product transportation route in Haikou area based on simulated annealing algorithm. Adv. Comput. Signals Syst. **5**(1), 71–74 (2021)
8. S. Xiao, X. Tan, J. Wang, A simulated annealing algorithm and grid map-based UAV coverage path planning method for 3D reconstruction. Electronics **10**(7), 853 (2021)
9. H. Miao, Y.-C. Tian, Dynamic robot path planning using an enhanced simulated annealing approach. Appl. Math. Comput. **222**, 420–437 (2013)
10. L. Wang, J. Guo, Q. Wang, J. Kan, Ground robot path planning based on simulated annealing genetic algorithm, in *2018 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC)*, pp. 417–4177 (2018). IEEE
11. U. Orozco-Rosas, O. Montiel, R. Sepúlveda, Mobile robot path planning using membrane evolutionary artificial potential field. Appl. Soft Comput. **77**, 236–251 (2019)
12. Z. Zhu, H. Lyu, J. Zhang, Y. Yin, An efficient ship automatic collision avoidance method based on modified artificial potential field. J. Mar. Sci. Eng. **10**(1), 3 (2021)
13. S. Feng, Y. Qian, Y. Wang, Collision avoidance method of autonomous vehicle based on improved artificial potential field algorithm. Proc. Inst. Mech. Eng. Part D J. Automobile Eng. **235**(14), 3416–3430 (2021)
14. A. Vagale, R. Oucheikh, R.T. Bye, O.L. Osen, T.I. Fossen, Path planning and collision avoidance for autonomous surface vehicles I: a review. J. Mar. Sci. Technol. 1–15 (2021)
15. W. Deng, J. Xu, H. Zhao, An improved ant colony optimization algorithm based on hybrid strategies for scheduling problem. IEEE Access **7**, 20281–20292 (2019)
16. L. Yue, H. Chen, Unmanned vehicle path planning using a novel ant colony algorithm. EURASIP J. Wirel. Commun. Netw. **2019**(1), 1–9 (2019)

17. Y. Su, J. Liu, X. Xiang, X. Zhang, A responsive ant colony optimization for large-scale dynamic vehicle routing problems via pheromone diversity enhancement. Complex Intell. Syst. **7**(5), 2543–2558 (2021)
18. S. Zhang, J. Pu, Y. Si, L. Sun, Path planning for mobile robot using an enhanced ant colony optimization and path geometric optimization. Int. J. Adv. Robot. Syst. **18**(3), 17298814211019222 (2021)
19. S. Katoch, S.S. Chauhan, V. Kumar, A review on genetic algorithm: past, present, and future. Multimedia Tools Appl. **80**(5), 8091–8126 (2021)
20. X. Sui, D. Liu, L. Li, H. Wang, H. Yang, Virtual machine scheduling strategy based on machine learning algorithms for load balancing. EURASIP J. Wirel. Commun. Netw. **2019**(1), 1–16 (2019)
21. Y.V. Pehlivanoglu, P. Pehlivanoglu, An enhanced genetic algorithm for path planning of autonomous UAV in target coverage problems. Appl. Soft Comput. **112**, 107796 (2021)
22. N.A. Shiltagh, K.S. Ismail, Z.Q. Habeeb, A modified genetic algorithm path planning for intelligent autonomous mobile robot. Invent. Rapid Algorithm (2012)
23. C. Li, W. Li, J. Ning, Calculation of ship collision risk index based on adaptive fuzzy neural network, in *Proceddings of the 2018 3rd International Conference on Modeling, Simulation and Applied Mathematics (MSAM 2018)*, vol. 160, pp. 223–227 (2018)
24. J. Ning, H. Chen, T. Li, W. Li, C. Li, Colregs-compliant unmanned surface vehicles collision avoidance based on multi-objective genetic algorithm. IEEE Access **8**, 190367–190377 (2020)
25. V. François-Lavet, P. Henderson, R. Islam, M.G. Bellemare, J. Pineau et al., An introduction to deep reinforcement learning. Found. Trends® Mach. Learn. **11**(3–4), 219–354 (2018)
26. Z. Chen, X. Wang, Decentralized computation offloading for multi-user mobile edge computing: a deep reinforcement learning approach. EURASIP J. Wirel. Commun. Netw. **2020**(1), 1–21 (2020)
27. H. Shen, H. Hashimoto, A. Matsuda, Y. Taniguchi, D. Terada, Automatic collision avoidance of ships in congested area based on deep reinforcement learning, in *Conference Proceedings, the Japan Society of Naval Architects and Ocean Engineers*, pp. 651–656 (2017)
28. L. Li, D. Wu, Y. Huang, Z.-M. Yuan, 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 (2021)
29. J. Gao, W. Ye, J. Guo, Z. Li, Deep reinforcement learning for indoor mobile robot path planning. Sensors **20**(19), 5493 (2020)
30. F. Duchoň, A. Babinec, M. Kajan, P. Beňo, M. Florek, T. Fico, L. Jurišica, Path planning with modified a star algorithm for a mobile robot. Procedia Eng. **96**, 59–69 (2014)
31. G. Tang, C. Tang, C. Claramunt, X. Hu, P. Zhou, Geometric a-star algorithm: an improved a-star algorithm for AGV path planning in a port environment. IEEE Access **9**, 59196–59210 (2021)
32. R. Song, Y. Liu, R. Bucknall, Smoothed a* algorithm for practical unmanned surface vehicle path planning. Appl. Ocean Res. **83**, 9–20 (2019)
33. B. Guo, Z. Kuang, J. Guan, M. Hu, L. Rao, X. Sun, An improved a-star algorithm for complete coverage path planning of unmanned ships. Int. J. Pattern Recognit. Artif. Intell. **36**(03), 2259009 (2022)
34. Y. Singh, S. Sharma, R. Sutton, D. Hatton, A. Khan, 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 (2018)
35. Y. Xu, J. Zhang, Y. Ren, Y. Zeng, J. Yuan, Z. Liu, L. Wang, D. Ou, Improved vessel trajectory prediction model based on stacked-bigrus. Secur. Commun. Netw. **2022** (2022)
36. S. Ahuja, N.A. Shelke, P.K. Singh, A deep learning framework using CNN and stacked bi-GRU for covid-19 predictions in India. SIViP **16**(3), 579–586 (2022)
37. T. Alfakih, M.M. Hassan, A. Gumaei, C. Savaglio, G. Fortino, Task offloading and resource allocation for mobile edge computing by deep reinforcement learning based on sarsa. IEEE Access **8**, 54074–54084 (2020)
38. R.S. Sutton, A.G. Barto, *Reinforcement Learning: An Introduction* (MIT Press, Cambridge, 2018)
39. B. Li, F.-W. Pang, An approach of vessel collision risk assessment based on the d-s evidence theory. Ocean Eng. **74**, 16–21 (2013)
40. E.M. Goodwin, A statistical study of ship domains. J. Navig. **28**(3), 328–344 (1975)
41. Y. Huang, L. Chen, P. Chen, R.R. Negenborn, P. Van Gelder, Ship collision avoidance methods: State-of-the-art. Saf. Sci. **121**, 451–473 (2020)
42. R. Zhen, M. Riveiro, Y. Jin, A novel analytic framework of real-time multi-vessel collision risk assessment for maritime traffic surveillance. Ocean Eng. **145**, 492–501 (2017)
43. Y. Yin, Y. Li, H. Gao, T. Liang, Q. Pan, FGC: GCN based federated learning approach for trust industrial service recommendation. IEEE Trans. Ind. Inform. (2022)
44. H. Gao, X. Qin, R.J.D. Barroso, W. Hussain, Y. Xu, Y. Yin, Collaborative learning-based industrial iot api recommendation for software-defined devices: the implicit knowledge discovery perspective. IEEE Trans. Emerg. Top. Comput. Intell. (2020)
45. H. Gao, K. Xu, M. Cao, J. Xiao, Q. Xu, Y. Yin, The deep features and attention mechanism-based method to dish healthcare under social IoT systems: an empirical study with a hand-deep local-global net. IEEE Trans. Comput. Soc. Syst. **9**(1), 336–347 (2021)
46. Y. Yin, Z. Cao, Y. Xu, H. Gao, R. Li, Z. Mai, QoS prediction for service recommendation with features learning in mobile edge computing environment. IEEE Trans. Cognit. Commun. Netw. **6**(4), 1136–1145 (2020)

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.