

Research Article

Research on Ship Meteorological Route Based on A-Star Algorithm

Ge Chen ¹, Tao Wu ^{1,2} and Zheng Zhou³

¹Chengdu University of Information Technology, School of Computer Science, Chengdu 610225, China

²Sichuan Numerical Weather Computing Engine Research Center, Chengdu 610225, China

³Ninecosmos Science and Technology Ltd., Wuxi 21400, China

Correspondence should be addressed to Tao Wu; wut@cuit.edu.cn

Received 28 March 2021; Accepted 30 April 2021; Published 18 May 2021

Academic Editor: Xianyong Li

Copyright © 2021 Ge Chen et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Ship meteorological navigation is based on hydrometeorological data of a certain time scale, considering the ship's motion characteristics and its own characteristics. First, we provide the best route for the ship and then use real-time local weather information to correct the route during the ship's navigation. It can also be expressed as follows: it is based on the hydrological and meteorological conditions of the ship during its voyage and the seakeeping characteristics of the ship itself, and the route planning method is used to select the best route for the ship. The best route is a balance between economy and safety, that is, based on ensuring the safety of ship navigation, the route that meets the shortest navigation time, the least fuel consumption, or the least navigation risk is obtained. Weather navigation includes the optimization of the initial route before sailing and the correction of the route after sailing. As there may be errors in hydrometeorological forecasts, especially in the accuracy and real-time performance of medium and long-term forecasts, the optimal initial route may not achieve the best results. Therefore, after the ship sails, it is necessary to adjust and correct the preferred initial route based on the meteorological information detected by the sensors or the continuously updated hydrometeorological forecast data to ensure the best effect of meteorological navigation. This paper proposes a weather route planning method based on the improved A-star algorithm. The convex shape of the concave obstacle and the expansion of the obstacle are carried out; according to the position of the target point relative to the starting point, the search direction of the A-star algorithm at each node is restricted, and an improved A-star algorithm is proposed. The simulation of global weather route planning shows that the improved A-star algorithm can not only find the optimal path but also effectively reduce the number of nodes that the algorithm needs to search during operation. Compared with the classic algorithm, the improved algorithm reduces the number of node searches by 29.25%.

1. Basic Theory and Methods of Meteorological Navigation

1.1. Overview of Meteorological Navigation. Ship meteorological navigation is based on the hydrometeorological data of a certain time scale, taking into account the ship's motion characteristics and its own characteristics. First, we provide the best route for the ship and then the navigation technology of correcting the route by using real-time local meteorological information in the course of ship navigation [1]. Meteorological navigation includes the selection of the initial route before sailing and the correction of the route after sailing. Due to the error of hydrometeorological forecast, especially the deviation of accuracy and real-time of medium and long-term

forecast, the optimal initial route may not achieve the best effect. Therefore, when the ship is sailing, it is necessary to adjust and correct the selected initial route according to the meteorological information detected by the sensor or the continuously updated hydrometeorological forecast data, to ensure the best effect of meteorological navigation.

1.2. Characteristics of Meteorological Navigation.

- (1) All hydrological and meteorological conditions are considered in the route planning before and after the voyage. Therefore, the meteorological navigation can prevent the ship from navigating in areas with heavy wind and waves to ensure the safety of the ship [2].

- (2) Navigation safety is the prerequisite for route planning, and weather navigation can plan different routes for ships with different sailing goals [2].

1.3. Presentation of Environmental Models. Cognitive map is a spatial model, which expresses the interconnection between objective things and their geographical distribution. Environmental modeling is the embodiment of the characteristics of the surrounding environment, and it is a cognitive map based on perceptual knowledge [3]. Most of the environmental modeling in route planning is to transform environmental information into maps, that is, to establish a network model. Common methods include grid-based map method [4], geometric feature map [5] method, and topological map method [6]. Montes [7] used a square grid map composed of longitude and latitude to divide the Western Pacific and established an *OSTR* (*optimum track ship routing*) model for the US Navy fleet. This paper also uses square grid graphs to represent the environment model. Since the research focus of this article is on the algorithm of route planning and does not consider the specific movement characteristics of the ship, this article simplifies the ship in a two-dimensional space and approximates it to a point to represent.

As shown in Figure 1, the ship's current position is $M(\lambda_i, \phi_i)$. Use the longitude lines from λ_1 to λ_n and the latitude lines from ϕ_1 to ϕ_n to divide the sea area containing the starting point $S(\lambda_1, \phi_1)$ and the target point $E(\lambda_n, \phi_n)$. The accuracy of the grid can be $0.5^\circ \times 0.5^\circ$, $1^\circ \times 1^\circ$, $2^\circ \times 2^\circ$, etc. A grid diagram is a directed network that can be defined as follows:

$$G = (N, A), \quad (1)$$

where N is the set of nodes and A is the set of directed arcs [8]. Each intersection of the warp and weft is a node. Suppose the accuracy of the square grid is $s^\circ \times s^\circ$, number the nodes in the order from bottom to top and from left to right, for example, the starting point S is numbered 1, and the current point of ship M is numbered $((\phi_n - \phi_1)/s + 1) \times (\lambda_i - \lambda_1)/s + ((\phi_i - \phi_1)/s + 1)$. As equation (2) and (3), if the node number is N_0 , then the latitude and longitude of the node has the following relationship with its number:

$$\lambda = \lambda_1 + \frac{N_0 - 1}{t} \times s, \quad (2)$$

$$\phi = \phi_1 + [(N_0 - 1) \bmod t] \times s. \quad (3)$$

Inside equation (3), $t = ((\phi_n - \phi_1)/s) + 1$.

The simulations in this article use a square grid graph with a moving direction, that is, each node has 8 out-degrees and 8 in-degrees [9], except for nodes on the border and corners, as shown in Figure 1. $T_1, T_2, T_3, T_4, T_5, T_6, T_7,$ and T_8 are the current node M where the ship is in order, starting from true north and clockwise at every 45° out of 8 directions and satisfying $T_1 = 0^\circ, T_2 = 45^\circ, T_3 = 90^\circ, T_4 = 135^\circ, T_5 = 180^\circ, T_6 = 225^\circ, T_7 = 270^\circ,$ and $T_8 = 315^\circ$. Due to the limitation of the boundary of the navigation area, the corner nodes have only 3 out-degrees and 3 in-degrees, and the nodes on the boundary have only 5 out-degrees and 5 in-degrees [10]. For

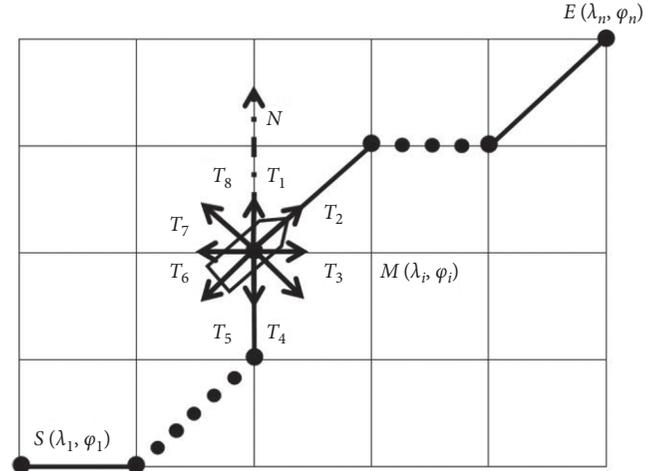


FIGURE 1: Grid-based map.

a node numbered N_0 with 8 out-degrees, the numbers of adjacent nodes in the 8 out-degree directions are $N_0 + 1, N_0 + t + 1, N_0 + t, N_0 + t - 1, N_0 - 1, N_0 - t - 1,$ and $N_0 - t + 1$. Inside, $t = ((\phi_n - \phi_1)/s) + 1$.

2. A-Star Algorithm Theoretical Basis

2.1. Dijkstra's Algorithm and Best First Search Algorithm. Dijkstra's algorithm visits the nodes in the graph from the starting point [11]. It constantly checks the nodes adjacent to the current node in the set of all nodes that have not found the shortest path and adds the node with the smallest cost value from the set of all nodes that have not found the shortest path to all the nodes that have found the shortest path. In the collection, it starts from the starting point and expands layer by layer until it finds the target point. If all edges in the graph have nonnegative cost values, Dijkstra's algorithm guarantees to find the shortest path from the starting point to the target point [12]. As shown in Figure 2, the green grid is the location of the starting node, the red grid is the location of the target node, the light blue area represents the area searched by Dijkstra algorithm, the light green grid represents the boundary of the search node area, and the light-yellow dashed line is the shortest path found by Dijkstra algorithm.

The steps of the best first search (BFS) algorithm are like the Dijkstra algorithm, and it can estimate the moving cost of the node to the target point. It favors nodes close to the target point rather than nodes close to the starting point. It runs fast, but there is no guarantee that the path found is the shortest [13]. The heuristic function used by the best first search algorithm can make it quickly approach the target point, and it runs faster than the Dijkstra algorithm under the same conditions. As shown in Figure 3, the green grid is the location of the starting node, the red grid is the location of the target node, the light blue area represents the area searched by best first search algorithm, the light green grid represents the boundary of the search node area, and the light-yellow dashed line is the shortest path found by best first search algorithm.

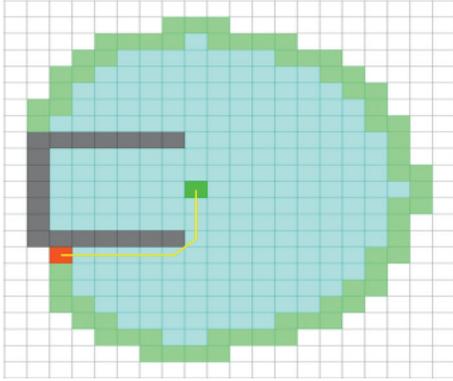


FIGURE 2: Route planning simulation of Dijkstra algorithm.

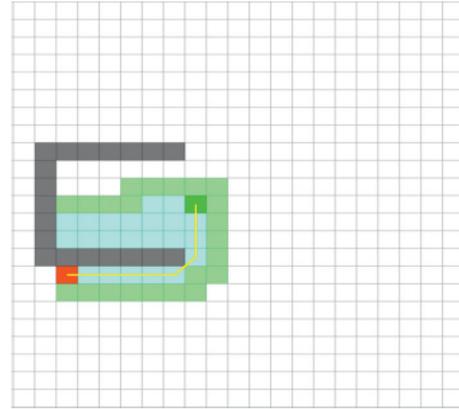


FIGURE 4: Route planning simulation of A-star algorithm.

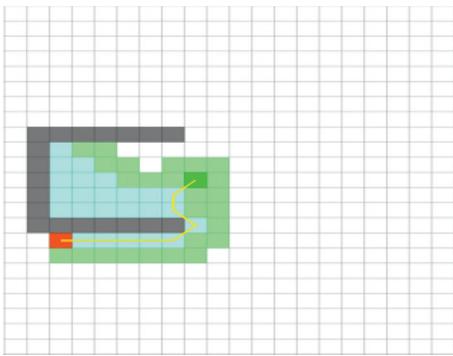


FIGURE 3: Route planning simulation of BFS algorithm.

By comparing Figures (2) and (3), we can find (1) Dijkstra algorithm can indeed find a shortest path; (2) the path found by the best first search algorithm is obviously not the shortest path; (3) the grid area scanned by the best first search algorithm is significantly less, so it runs faster than Dijkstra's algorithm.

2.2. A-Star Algorithm. The A-star algorithm concentrates the advantages of both the Dijkstra algorithm and the best first search algorithm. It is an algorithm that combines the advantages of the conventional algorithm (such as the Dijkstra algorithm) and the heuristic algorithm (such as the best first search algorithm) [14]. A-star algorithm is one of the most popular algorithms in path planning. It is quite flexible and can be used in various scenarios. Like the Dijkstra algorithm, the A-star algorithm can be used to find the shortest path [15]. Like the best first search algorithm, the A-star algorithm can use heuristic functions to guide the approach to the target point. In the simplest case, where there are no obstacles, the A-star algorithm can be as fast as the best first search algorithm. In the presence of concave obstacles as shown in Figures (2)–(4), the A-star algorithm can find the same good results as the Dijkstra algorithm. In Figure 4, the green grid is the location of the starting node, the red grid is the location of the target node, the light blue area represents the area searched by A-star algorithm, the light green grid represents the boundary of the search node area, and the light-yellow dashed line is the shortest path found by the A-star algorithm.

The A-star algorithm needs to use the cost function as follows:

$$F(n) = G(n) + H(n), \quad (4)$$

where $F(n)$ is the estimated value of moving from the starting point to the n th node, $G(n)$ is the cost value of moving from the starting point to the n th node along the generated path, and $H(n)$ is the heuristic function, used to estimate the cost value of moving from the n th node to the target point.

The A-star algorithm also needs to use the open table and closed table: the open table is used to store the nodes that have been.

The execution process of the A-star algorithm is as follows:

- (1) Add the starting point to the open table
- (2) Repeat the following tasks:
 - (1) Find the node with the lowest F value in the open table and set the current node as the node with the lowest F value.
 - (2) Remove the current node from the open table and put it into the closed table.
 - (3) Judge all nodes adjacent to the current node in order. (c1) If the node is already in the closed table, then the node is not considered. (c2) If the node is not in the open table, add the node to the open table and use the current node as the node the parent node of the node and record the value of F , G , and H of the node. (c3) If the node is already in the open table, the value of G that reaches the node after passing through the current node is less than the value of G that reaches the node without passing through the current node; the current node is taken as the parent node of the node, and the F , G , and H values of the node are recalculated. Otherwise, the F , G , and H values of the node remain unchanged.
 - (4) Stop searching: if the target point is already added to the closed table, the path is saved; if the target point is not added to the closed table and

the open table is empty, the path does not exist generated but not visited, and the closed table is used to store the nodes that have been visited.

2.3. The Influence of Heuristic Function $H(n)$ on A-Star Algorithm. Whether the A-star algorithm can guarantee to find the optimal solution, the key lies in the selection of the heuristic function $H(n)$. For the evaluation function: $F(n) = G(n) + H(n)$, the selection of the heuristic function $H(n)$ has the following situations:

- (1) In a limit case, if $H(n) = 0$, then $F(n) = G(n)$, that is, only $G(n)$ works. At this time, the A-star algorithm is equivalent to the Dijkstra algorithm, and the path found is the shortest path.
- (2) If $H(n)$ is always less than the cost of moving from node n to the target point, then the path found by the A-star algorithm is also the shortest path. Moreover, the smaller the value of $H(n)$, the more nodes the A-star algorithm expands, and the slower the execution speed of the A-star algorithm.
- (3) If $H(n)$ is exactly equal to the cost of moving from node n to the target point, then the A-star algorithm will only move along the optimal path without expanding other irrelevant nodes. At this time, the A-star algorithm will run very fast. Although this cannot happen in all scenarios, it can happen in some special scenarios.
- (4) If $H(n)$ is sometimes more expensive than moving from node n to the target point, the A-star algorithm cannot guarantee to find a shortest path, but it runs faster.
- (5) In another extreme case, if $H(n)$ is much larger than $G(n)$, that is, $F(n) \approx H(n)$, only $H(n)$ will work. At this time, the A-star algorithm will approximate for the best first search algorithm, and this situation cannot guarantee to find a shortest path, but the algorithm runs fast.

Ideally, we want to get the shortest path the fastest. If the value of $H(n)$ is too small, then we can get the shortest path, but the running speed of the algorithm will become very slow. If the value of $H(n)$ is too large, then we must abandon the shortest path, but the algorithm will run faster. We need to make a trade-off between the speed of the algorithm and the accuracy of the results. In some application scenarios, such as road strength planning in games, this feature of the A-star algorithm is very useful. Because it is not always necessary to obtain the optimal path, for example, the object in the game moves in a safe area and sometimes an approximate optimal solution can be used instead. At this time, you can appropriately sacrifice the accuracy of the algorithm's running results in exchange for the algorithm's running speed.

3. Research on Global Weather Route Planning Method Based on Improved A-Star Algorithm

Global route planning is static planning based on complete prior information. This simulation is using the improved A-star algorithm proposed in this paper to plan global routes for ships. The raster model is used to model the environment of the navigational sea area, and the concave obstacles are convexed and expanded. The simulation proves the efficiency of the improved A-star algorithm proposed in this chapter.

3.1. Selection of Numerical Weather Forecast Information. Wave height is one of the main meteorological factors affecting the navigation of ships. Therefore, the global route planning in this chapter only considers the wave height information in the numerical weather forecast information. In addition, the research in this chapter focuses on the improvement of the A-star algorithm, and for the sake of simplicity, this chapter identifies sea areas with wave heights higher than 6 m as nonnavigable areas like static obstacles (shallow waters, islands, reefs, etc.), and directly the sea area with wave height less than or equal to 6 m [16] is marked as a navigable area that is unobstructed to the navigation of ships.

3.2. Environment Modeling Based on Grid. Because the grid model has a strong ability to express the environment, especially for obstacles with irregular contours and the grid model is simple and easy to operate, this chapter uses the grid model to model the environment. As shown in Figure 5, the starting position of the ship is S , which is the green dot in Figure 5; the target position is E , which is the red dot in the Figure 5; the black shaded area is the static obstacle; and the gray shade is the coverage area is the nonnavigable area formed by meteorological information and is also regarded as a static obstacle.

3.3. Convex Type of Concave Obstacle. As shown in Figure 5, the static obstacles in the upper left and lower left corners are concave obstacles. If the target position is not in the concave area of the concave obstacle, the algorithm's search in the concave area is meaningless. Moreover, the concave area of the concave obstacle may cause some route planning algorithms to fall into local oscillation. Therefore, the concave obstacle needs to be convexed [17]. At the same time, the convex shape of the concave obstacle is also necessary for the improved A-star algorithm proposed in this chapter. As shown in Figure 6, two straight lines are, respectively, tangent to the notches of the obstacle in the upper left corner and the lower left corner to form the line segments ab and cd . The concave obstacle in the upper left corner becomes a convex obstacle surrounded by the line segment ab and the

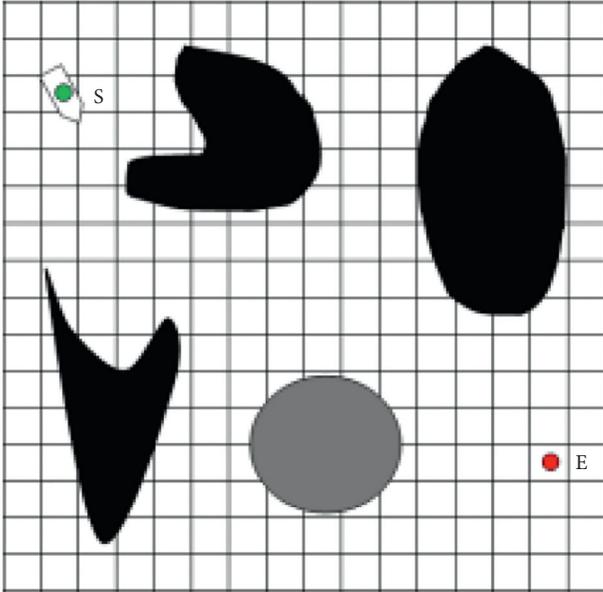


FIGURE 5: Grid model.

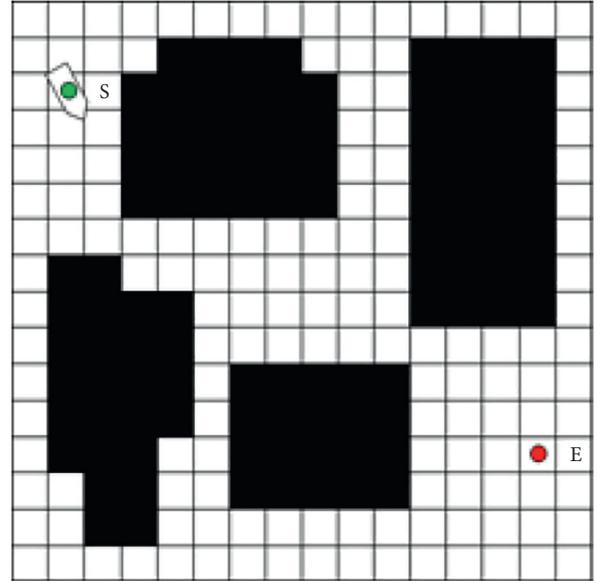


FIGURE 7: The expansion of obstacles.

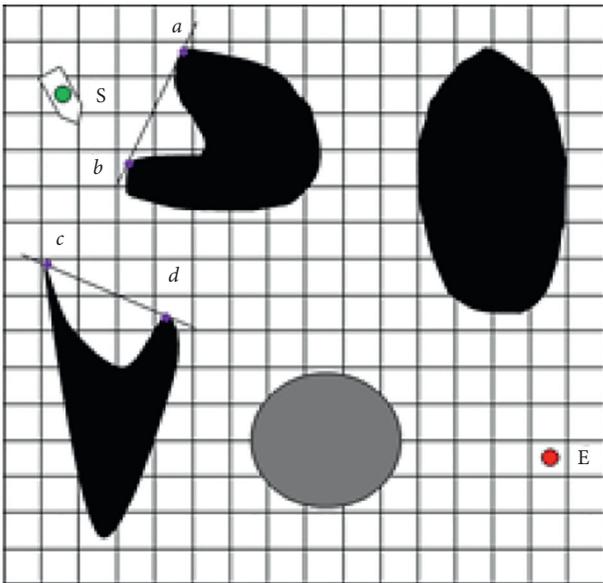


FIGURE 6: Convex type of concave obstacle.

original obstacle in the upper left corner. The concave obstacle in the lower left corner becomes a convex obstacle surrounded by the line segment cd and the original obstacle in the lower left corner.

3.4. The Expansion of Obstacles. As shown in Figure 7, to simplify the problem, the ship is represented by a point, and at the same time, the outline of the obstacle needs to be extended a certain distance. The expansion process in this chapter is based on the convexization of concave obstacles. It is considered that if the grid contains convex obstacles (including convex obstacles formed by the convexity of concave obstacles and those that are not convexized), convex

barrier is a part of any size, the grid is a nonnavigable area. Because the contour of the convex obstacle generally does not fill the grid, the expansion treatment will also form a certain length of safety distance.

3.5. A-Star Algorithm Improvement. The A-star algorithm is obtained by introducing a heuristic function based on the Dijkstra algorithm, so the classic A-star algorithm is an improved Dijkstra algorithm. Although the classic A-star algorithm has been improved, it still has its shortcomings [18]. For example, in the process of raster map route planning, when the classic A-star algorithm is used, it is necessary to traverse all feasible nodes adjacent to each node. Once again, when the number of nodes in the raster map is large, the calculation amount of the classic A-star algorithm to find the optimal path will increase sharply, resulting in a decrease in the efficiency of finding the optimal path [19]. The improvement of the A-star algorithm in this chapter is as follows: according to the 0020 position of the target point relative to the starting point, the algorithm's search direction at each node is restricted. Taking a square grid map with 8 search directions as an example, after the preprocessing of the global route planning, the specific improvement process of the A-star algorithm is as follows:

- (1) Connect the starting point and the target point with a straight line and measure the angle α between the line and the true north direction. The calculation method of α is as follows:

$$\alpha = \omega \arctan\left(\frac{|\lambda_e - \lambda_s|}{|\phi_e - \phi_s|}\right) + \theta. \quad (5)$$

When $(\lambda_e - \lambda_s) > 0$ and $(\phi_e - \phi_s) > 0$, then $\theta = 0^\circ$ and $\omega = 1$.

TABLE 1: Summary table of the relationship between the α and the direction of the search out-degree.

α	Reserve 5 search out-degree directions	Discard 3 search out-degree directions
$[337.5^\circ, 360^\circ) \cup [337, 5^\circ, 360^\circ)$	000T, 045T, 090T, 270T, 315T	135T, 180T, 225T
$[22.5^\circ, 67.5^\circ)$	000T, 045T, 090T, 135T, 315T	180T, 225T, 270T
$[67.5^\circ, 112.5^\circ)$	000T, 045T, 090T, 135T, 180T	225T, 270T, 315T
$[112.5^\circ, 157.5^\circ)$	045T, 090T, 135T, 180T, 225T	270T, 315T, 000T
$[157.5^\circ, 202.5^\circ)$	090T, 135T, 180T, 225T, 270T	000T, 045T, 315T
$[202.5^\circ, 247.5^\circ)$	135T, 180T, 225T, 270T, 315T	000T, 045T, 090T
$[247.5^\circ, 292.5^\circ)$	180T, 225T, 270T, 315T, 000T	045T, 090T, 135T
$[292.5^\circ, 337.5^\circ)$	225T, 270T, 315T, 000T, 045T	090T, 135T, 180T

TABLE 2: Comparison of the number of node searches between the classic A-star algorithm and the improved A-star algorithm.

Algorithm	Number of node searches	Percentage reduction in the number of node searches (%)
Classic A-star algorithm	106	—
Improved A-star algorithm	75	29.25

When $(\lambda_e - \lambda_s) > 0$ and $(\varphi_e - \varphi_s) < 0$, then $\theta = 180^\circ$ and $\omega = -1$.

When $(\lambda_e - \lambda_s) < 0$ and $(\varphi_e - \varphi_s) < 0$, then $\theta = 180^\circ$ and $\omega = 1$.

When $(\lambda_e - \lambda_s) < 0$ and $(\varphi_e - \varphi_s) > 0$, then $\theta = 360^\circ$ and $\omega = -1$.

Among them, λ_e is the longitude of the target point, λ_s is the longitude of the starting point, φ_e is the latitude of the target point, φ_s is the latitude of the starting point, and θ and ω are constants.

- (2) As shown in Table 1, according to the relationship between the α and the search out degree direction, determine the reserved search out degree direction.
- (3) The included angle α is introduced as heuristic information into the A-star algorithm to find the optimal path:

- (1) Add the starting point to the open table:
 - (1) Find the node with the lowest F value in the open table and set the current node to the node with the lowest F value.
 - (2) Move the current node out of the open table and put it into the closed table.
 - (3) Determine the 5 nodes pointed to by the reserved 5 search directions in order. (c1) If the node is already in the closed table, then ignore the node. (c2) If the node is not in the open table, then add the node to the open table and use the current node as the parent node of the node and record the F , G , and H values of the node. (c3) If the node is already in the open table, when the node is reached through the current node, and if the value of G is less than the value of G that reached the node without passing through the current node, the current node is taken as the parent node of the node and the F , G , and H values of the node are recalculated. Otherwise, the F , G , and H of the node the value remain unchanged.
 - (4) Stop searching: if the target point has been added to the closed table, save the path; if the target

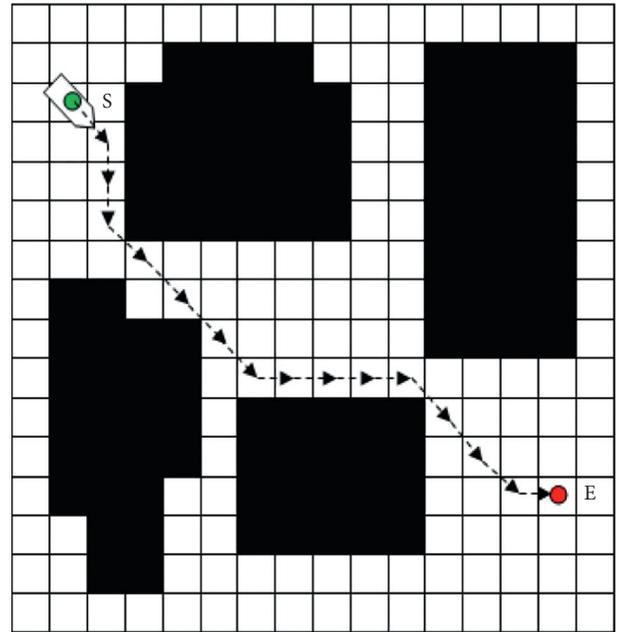


FIGURE 8: Classic A-star algorithm simulation result graph.

point is not added to the closed table and the open table is empty, the path does not exist.

4. Simulation of Global Weather Route Planning Based on Improved A-Star Algorithm

4.1. Simulation content. For the raster map shown in Figure 7, use the classic A-star algorithm and the improved A-star algorithm in this chapter from the starting point S (121.5°E , 38.5°N) to the target point E (134.5°E , 28.5°N). Plan the optimal route and count the number of nodes searched during the route planning process.

For the improved A-star algorithm, according to Equation 5, we can get

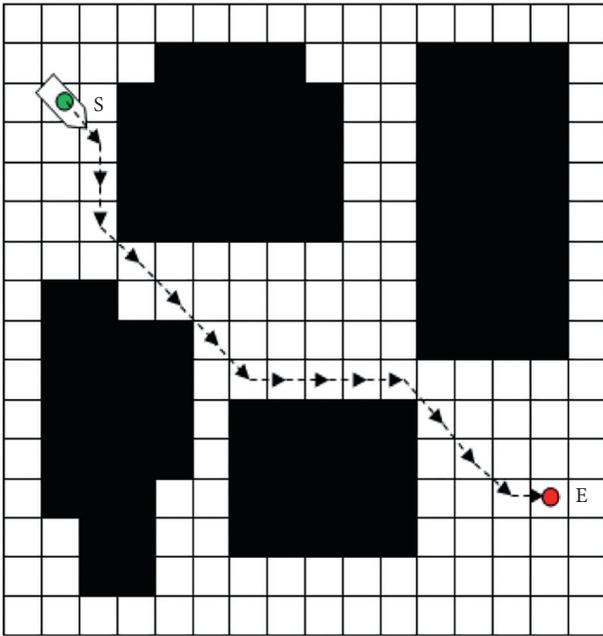


FIGURE 9: Improved A-star algorithm: algorithm simulation result graph.

$$\begin{aligned}\alpha &= (-1) \times \arctan\left(\frac{134.5 - 121.5}{|28.5 - 38.5|}\right) + 180^\circ \\ &= 127.57^\circ \in [112.5^\circ, 157.5^\circ).\end{aligned}\quad (6)$$

Combined with Table 1, according to the included angle α , the five search out-degree directions that need to be retained are 045T, 090T, 135T, 180T, and 225T.

Therefore, the search direction of each node of the improved A-star algorithm is these 5 directions.

4.2. Simulation Results and Analysis. As shown in Table 2 and Figures 8 and 9, both the classic A-star algorithm and the improved A-star algorithm can find the optimal path. In Figures 8 and 9, compared with the classic A-star algorithm, the improved A-star algorithm reduces the number of node searches by 29.25%, so the improved A-star algorithm proposed in this chapter is feasible and efficient.

5. Conclusion

This paper proposes an improved A-star algorithm for ship weather route planning, in areas where the wave height does not exceed 6 meters and considering static obstacles, the algorithm can find the shortest path, compared with the traditional A* algorithm, it reduces the number of search nodes by 29.5%, the time-consuming calculation of the route is reduced, and the efficiency of the algorithm is improved to a certain extent, which can well meet the real-time requirements in the meteorological route planning.

Data Availability

The data used to support the study are not publicly available.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was funded by the National Key Research (number 2018YFC15007005) and Key R&D projects in Sichuan Province (number 2020YFG0189).

References

- [1] Lizhihua and Wanghui, *Marine Weather Navigation*, Dalian Maritime University Press, Dalian, China, (in Chinese), 2006.
- [2] Yanglianghua, "Introduction to ship weather navigation," *Meteorological*, vol. 2, no. 7, pp. 19-20, 1982, (in Chinese).
- [3] Tangqinghui, *Research on Route Planning Method Based on Electronic Chart*, Ocean University of China, Qingdao, China, (in Chinese), 2011.
- [4] H. P. Moravec and A. Elfes, "High resolution maps from angle sonar," in *Proceedings of the International Conference on Robotics Research*, pp. 116-121, St. Louis, MO, USA, March 1985.
- [5] R. Chatila and J. Laumond, "Position referencing and consistent world modeling for mobile robots," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 138-145, Hong Kong, China, March 2014.
- [6] J. C. Latombe, "Robot motion planning with uncertainty in control and sensing," *Artificial Intelligence*, vol. 52, no. 52, p. 52, 1991.
- [7] A. A. Montes, "Network shortest path application for optimum track ship routing," M. S. Thesis, Operations Research Department Naval Postgraduate School, Monterey, CA, USA, 2005.
- [8] C. Huxiaomei and Y. Zhaotao, "Summary of path planning algorithm and its application," *Guizhou: Modern Machinery*, vol. 5, pp. 85-90, 2011, (in Chinese).
- [9] R. A. Brooks, *Solving the Find-Path Problem by Good Representation of Free Space. Autonomous Robot Vehicles*, pp. 190-197, Springer, New York, NY, USA, 1990.
- [10] J. Wangfengwu, *The Study on the Optimal Ship routing*, no. 2, Dalian Maritime University Press, (in Chinese), pp. 61-64, 1998.
- [11] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, no. 1, pp. 269-271, 1959.
- [12] Y. Wang, L. Zhou, B. Yu et al., "3D auto-context-based locality adaptive multi-modality GANs for PET synthesis," *IEEE Transactions on Medical Imaging*, vol. 38, no. 6, pp. 1328-1339, 2019.
- [13] D. Harabor and A. Grastien, "The JPS pathfinding system," in *Proceedings of the Fifth Annual Symposium on Combinatorial Search*, Ontario, Canada, July 2012.
- [14] P. E. Hart, N. J. Nilsson, and B. Raphael, "Correction to A Formal basis for the heuristic determination of minimum cost paths," *IEEE Transactions on Systems Science & Cybernetics*, vol. 4, no. 2, pp. 100-107, 2007.
- [15] S. Koenig and M. Likhachev, "Fast replanning for navigation in unknown terrain," *IEEE Transactions on Robotics*, vol. 21, no. 3, pp. 354-363, 2005.
- [16] C. Liyuanlin, "Design of optimum ship route using weather routing techniques," *Journal of South China University of Technology (Natural Science)*, vol. 3, 1997.

- [17] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 90–98, Taipei, Taiwan, September 2003.
- [18] M. Steinbrunn, G. Moerkotte, and A. Kemper, "Heuristic and randomized optimization for the join ordering problem," *The VLDB Journal*, vol. 6, 1997.
- [19] P. Tang, C. Zu, M. Hong et al., "DA-DSUnet: dual Attention-based Dense SU-net for automatic head-and-neck tumor segmentation in MRI images," *Neurocomputing*, vol. 435, pp. 103–113, 2021.