Long-voyage route planning method based on multi-scale Visibility Graph for autonomous ships

Gongxing Wu [a]*, Incecik Atilla [b], Tezdogan Tahsin [b], Momchil Terziev [b], LingChao Wang [a]

[a] College of Ocean Science and Engineering, Shanghai Maritime University, Shanghai, 201306, China

[b] Department of Naval Architecture, Ocean and Marine Engineering, University of Strathclyde, Glasgow, UK

Abstract: With the increasing demand for the autonomous ship, a fast planning method for long-distance ship routes is needed. In this paper, a multi-scale Visibility Graph(VG) method is proposed for long-voyage route planning, as a solution to the problems of the slow planning and poor route accuracy. First, polygon data of obstacles are extracted from an electronic chart. In order to reduce the number of Visibility Points(VPs), the VPs are expanded from the convex points of these polygons. The small-scale, medium-scale, and large-scale VG models are established respectively. Second, this paper proposes the Local Planning Window(LPW) method, which greatly reduces the complexity of the VG models. The great circle route method is used to decompose the longer route, which further shorten the search time of the VG. The route planning process is designed for the multi-scale VG method. Finally, a long-voyage route planning example is carried out, in which, the utilization rate of the number of obstacle polygons and the number of VPs are analyzed. The data results show that: the complexity of VG models can be reduced greatly, and the search time of the VG will be shortened, by using the multi-scale VG method.

Keywords: ship route planning; multi-scale map; visibility graph; autonomous ship

1. Introduction

Autonomous ships seem to be accepted by the marine community, primarily for economic reasons, and will continue being developed. Kongsberg and Rolls-Royce had designed two autonomous ships which will be tested on the sea routes from the Baltic corridor to the largest ports of the European Union(Felski and Zwolak, 2020). The shipping 4.0 has already started and is gradually beginning to transform the design, building, operation, maintenance and manning of ships. In 2017, the IMO adopted the term Maritime Autonomous Surface Ships (MASS) to refer to ships of future whether unmanned or fully autonomous(Emad et al., 2020). The design of autonomous ships and shipping 4.0 are current research hotspots, and the automatic route planning is one of the key scientific issues requiring further investigation. Ocean-going ships have long sailing times and distances. Therefore, to shorten the sailing distance and reduce sailing cost is especially important for ocean-going ships. Ship route planning can not only ensure the safety of ship navigation, achieve energy savings by emission reduction, protect the environment by reducing pollution, and reduce operating costs, but also provide auxiliary decision-making for seafarers and reduce their workload. Therefore, research on ship route intelligence planning has important theoretical research significance for the realization of intelligent ships and shipping 4.0.

There are many methods for ship route planning. Depending on the different theories, the ship route planning methods can be divided into the following three categories: traditional method, big data mining

*E-mail: wugx@shmtu.edu.cn

method, and graphics method. But also there are other methods, such as particle swarm optimization for dynamic routing(Lin et al., 2017; Lin et al., 2018; Lin, 2018), genetic algorithms(Miao et al., 2017), and artificial potential field(Lee et al., 2019).

Ancient ocean-going voyages generally used climatic routes. This method is based on a large number of meteorological literature and summarizes the precious sailing experience of predecessors. This isochronous method is a simple manual route planning method proposed by Hanssen and James (1960). It defines the boundary of the area that can be reached after a certain time. This method was improved to make it suitable for computer programming by Hagiwara (1991). The application of variation algorithm to route planning with the least fuel consumption was proposed by Bijlsma (2008), and Kosmas (2008). The ship route planning problem can be seen as an optimization problem. To solve this, a method for a simultaneous determination of the path and the speed of a ship is proposed(Lee et al., 2018). Szlapczynska and Szlapczynski (2019) present a new tradeoff-based evolutionary multi-objective optimization approach utilizing configurable weight intervals assigned to all objectives. The proposed method has been applied to ship weather routing problem and compared with a popular reference point method, known as r-dominance. A generic extendable framework is proposed to provide the optimal route from multiple route planning objectives. These objectives are attained by an evaluation of multi-source input data, including a state-of-the-art model data for ice conditions, bathymetric knowledge, and ship-ice interaction(Lehtola et al., 2019). Sen and Padhy (2015), presented a general approach for the development of a ship weather routing algorithm for determining optimal route which is taken here as the minimum-time route. To reduce fuel cost, a method for determining an economical route for a ship based on the acquisition of the sea state and estimation of fuel consumption is proposed (Roh,2013). In the traditional method, the choice of this route depends largely on the experience accumulated by the captain, and is greatly influenced by individuals. For this reason, the safety and economy of ship navigation are restricted to a certain extent when using the above mentioned method(Zhou,2016).

The big data mining method is the extraction of patterns and knowledge from large amounts of ship route data. A data-driven model based on neural network theory is developed to predict the energy efficiency of ships in the Arctic. A method was proposed to generate shipping routes based on historical ship tracks. The ship's historical route information was obtained by processing Automatic Identification System(AIS) data(Zhang et al., 2019). An Artificial Neural Network (ANN) for automatic ship route was designed based on massive AIS data between certain ports(Wen et al., 2020). An approach was studied that AIS based shipping routes using the Dijkstra Algorithm in (Silveira et al., 2019). The big data mining method requires a large number of ship routes. If there is too little data in a certain area, the optimal ship route cannot be obtained.

The graphics method is an environment modelling to ship route planning. The grid method was first proposed by Moravec and Elfes (1985). A rectangular grid pattern was used to divide the Western Pacific and established the optimum track ship routing model for the US Navy fleet(Montes,2005). A path planning algorithm integrating a Voronoi diagram, Visibility algorithm, and Dijkstra search algorithm was proposed by Niu et al. (2016). In Candeloro et al. (2017) a ship routing method was proposed which is based on the Voronoi diagram and generates the initial path while ensuring that clearance constraints are satisfied with respect to both land and shallow waters. Visibility graphs are very useful in determining the shortest path(Kaluđer,2011). A method was proposed to compute optimal paths using A* search on visibility graphs defined over quadtrees in Shah and Gupta (2016). In this approach, specific geographical terrain is converted into a two-dimensional polygon-shaped environment. A trajectory is then generated using visibility graph method in (Kulbiej,2018). The visibility graph method is suitable for environment modelling of global route plan.

In summary, the referenced ship route planning studies were based on single-map, single-scale electronic charts. However, in long distance navigation, multiple charts are needed. If a long-distance route is generated by a small-scale chart, due to insufficient representation of chart details, the planned route will have lower accuracy in local areas and reduce ship safety. If it is generated by a large-scale charts, due to the large amount of chart data and associated information, the route planning time will be very long, which is not practical. Therefore, it is necessary to study data modelling based on multi-map and multi-scale charts, and the long-distance route planning method for autonomous ships based on this model. For these new problems, in this paper, a multi-scale visibility graph method is proposed for long-voyage route planning. The main contributions of this paper are summarized as follows:

(1) In order to reduce the number of Visibility Points, this paper proposes to use only the convex points of the obstacle polygon to build the Visibility Graph model. So the number of Visibility Points can be reduced nearly a half, and it will reduce the search time of Visibility Graph.

(2) The proposed Local Planning Window method is that only those Visibility Points near the route are selected for the Visibility Graph, which will reduce a large number of Visibility Points, thereby saving planning time.

(3) Then a new multi-scale Visibility Graph method is proposed for long-voyage route planning, in which, the small-scale Visibility Graph model is used to plan the rough route, while medium-scale and large-scale Visibility Graph models are used to refine the route. This approach can be a solution to the problems of the slow planning and poor route accuracy.

The rest of the paper is organized as follows. In Section 2, the Visibility Graph modelling method based on electronic charts is introduced, including obstacle polygon data extraction, polygon expansion algorithm and Visibility Points extraction algorithm. Section 3 focuses on the modelling of multi-scale Visibility Graph, and simplifying algorithms of Visibility Graph. The Local Planning Windows method is proposed in Section 4, and the route planning method based on multi-scale Visibility Graph models is introduced, and how to use the great circle route method to decompose longer routes. The long-distance route planning for intelligent ship is described in Section 5, and the planning results are analyzed; In Section 6, the application of multi-scale Visibility Graph method to long-distance route planning is summarized, and the contributions of the paper are presented.

2. Visibility Graph modelling method based on electronic chart

The research of electronic chart display and information system (ECDIS) began as early as the 1990s. After more than ten years of development, it has now entered the application stage. Automatic route planning can be realized by using the marine environment information provided by electronic charts, which can not only improve the safety of ship navigation, but also reduce shipping costs. Electronic charts are data models that describe geographic information. The model uses a combination of plane geometry, topological relationship and data structures to represent geographic information. The chart is divided into 19 layers, such as measurement control points, land objects, landforms, water systems, residential areas, transportation, pipelines and walls, ocean and land, water depth and bottom quality, port facilities, navigation aids , obstacles, offshore facilities, waterways, regional boundaries, service facilities, hydrological and magnetic elements, map indices and data files.

2.1. Extracting obstacle polygons from electronic charts

According to the manual of the electronic charts, chart layer information related to the route planning is found out, such as ocean and land, water depth and bottom quality, port facilities, navigation aids, obstacles, offshore facilities, waterways. The obstacles polygon data of land and islands are extracted from the ocean and land layers. The polygon data of shallow water areas of depth 0-10m are extracted by comparing with the water depth data. Other smaller obstacle data are not considered here, because this paper is focused on the long distance route planning. Using the ocean and land layer data, the obstacles polygon data of land, islands, and 0-10m shallow water areas could be extracted by analyzing the electronic charts data structure.

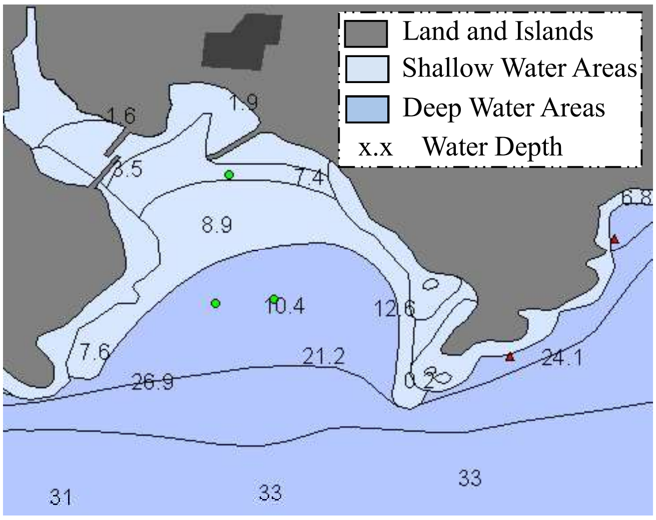


Fig. 1. The information display of electronic chart

As shown in Fig. 1, an electronic chart of a small port is shown, where the colors of ground and islands are gray, the 0-10m shallow water area color is grayish white, and the light blue is the deep water area.

2.2. Expansion algorithm for the polygon of obstacles

The ship is generally treated as a point, when modelling the marine environment. But the actual ship is an object with a length, width and height. Therefore, when modelling the chart, the actual size of the ship needs to be considered. In addition, due to the large mass of the ship, its motion inertia is also relatively large. In this paper, the method of expanding the obstacles polygon is adopted to make the modeled obstacles polygon larger than the actual polygon. This will keep the ship away from land and other obstacles to ensure the safety of the ship.

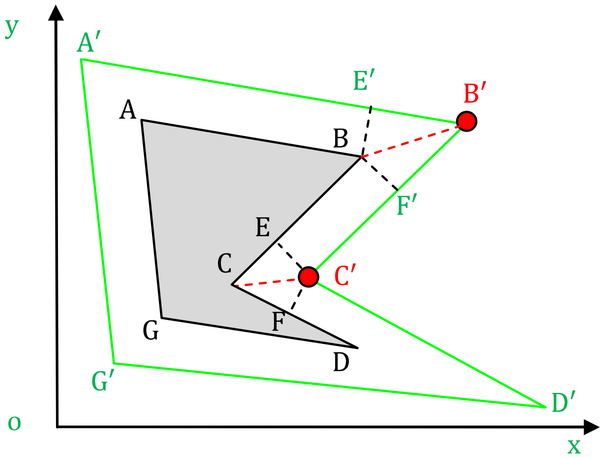


Fig. 2 Geometric diagram of chart expansion method

The polygon data has been obtained previously by using the ocean and land layer data. The expansion algorithm will be discussed here. As showed in Fig. 2, the polygon ABCDG will be expanded out by the distance d，and it will become the polygon A'B'C'D'G'. Given the vertex coordinates of polygon ABCDG and the expansion distance d, we now need to solve the vertex coordinates of polygon A'B'C'D'G' after expansion. Polygons have two types of vertices, convex and concave. As shown in Fig. 2, B is convex and C is concave. First, Fig. 2 shows how to solve for the corresponding point B' of the convex point B. The line BE' is the vertical line from point B to line A'B'. The line BF' is the vertical line from point B to line B'C'. From the expansion, the length of line BE' and the line BF' is the expansion distance d, then BB' is the angle bisector of angle A'B'C'. The formula of solving the coordinates of the vertex B ' is as follows:

$$\begin{cases} P_{B'}.x = P_B.x + L_{BB'} * \sin(\varphi_{BB'}) \\ P_{B'}.y = P_B.y + L_{BB'} * \cos(\varphi_{BB'}) \end{cases} \tag{1}$$

In which, $\mathrm{L_{BB'}} = d/\sin\,(0.5 * \angle\mathrm{ABC})$. When the angle $\angle\mathrm{ABC}$ becomes smaller, the length $\mathrm{L_{BB'}}$ becomes longer. So, this length is limited to three times the length of d in this paper. That is $\mathrm{L_{BB'}} \leq 3d.$ $\varphi_{BB'} = \varphi_{AB} - 0.5 * \angle\mathrm{ABC}.$ The d is the distance that the obstacles polygon be expanded. $\varphi_{BB'}$ is the vector direction of line BB'. The angle $\angle\mathrm{ABC}$ and the vector direction φ_{AB} can be calculated by the vertex coordinates of polygon ABCDG, as shown below:

$$\begin{cases} \angle ABC = \varphi_{BC} - \varphi_{AB} \\ \varphi_{AB} = \mathrm{atan2}(P_B.x - P_A.x, P_B.y - P_A.y) \\ \varphi_{BC} = \mathrm{atan2}(P_C.x - P_B.x, P_C.y - P_B.y) \end{cases} \tag{2}$$

In which, φ_{AB} is the vector direction of line AB. φ_{BC} is the vector direction of line BC. Then consider the coordinate calculation of the expansion point C 'of the concave point C. According to Fig. 2, the expansion points at the convex point B is compared with the concave point C. Only the calculation of angle $\angle$ABC is different in equation (1) and (2). The calculation of angle $\angle$BCD is as follows:

$$\begin{cases} \angle BCD = 2 * \pi - (\varphi_{CD} - \varphi_{BC}) \\ \varphi_{BC} = \mathrm{atan2}(P_C.x - P_B.x, P_C.y - P_B.y) \\ \varphi_{CD} = \mathrm{atan2}(P_D.x - P_C.x, P_D.y - P_C.y) \end{cases} \tag{3}$$

It is also necessary to determine whether the vertex of the polygon is a concave point or convex point by the cross product formula. Assume that the current three consecutive vertices of the polygon are $\mathrm{P_B}$，$\mathrm{P_C}$，$\mathrm{P_D}$。The formula for determining whether the vertex $\mathrm{P_C}$ is concave or convex is as follows:

$$\begin{cases} if\ C_{P_C} > 0\,, \quad when\ P_C\ is\ convex\ \ point \\ if\ C_{P_C} \leq 0, \quad when\ P_C\ is\ concave\ point \end{cases} \tag{4}$$

In which, $\mathrm{C_{P_C}}$ is the cross product of the vector in direction of $\varphi_{\mathrm{P_B P_C}}$ and $\varphi_{\mathrm{P_C P_D}}$. The corresponding equations are:

$$\begin{cases} C_{P_C} = \varphi_{P_B P_C}.x * \varphi_{P_C P_D}.y - \varphi_{P_B P_C}.y * \varphi_{P_C P_D}.x \\ \varphi_{P_B P_C}.x = P_B.x - P_C.x \\ \varphi_{P_B P_C}.y = P_B.y - P_C.y \\ \varphi_{P_C P_D}.x = P_C.x - P_D.x \\ \varphi_{P_C P_D}.y = P_C.y - P_D.y \end{cases} \tag{5}$$

As shown in Fig. 4, the green polygon represents the expanded polygon of the obstacle, and the obstacle is represented by gray as the non-navigable area. In Visibility Graph modelling, whether two points are visible is judged by whether the line connecting these two points intersects these expanded polygons. When performing the route planning, it is necessary to check whether the optimized route intersects these expanded polygons.

2.3. Visibility Graph modelling

After the expansion of the obstacle polygon, this section will discuss how to establish a Visibility Graph model. The Visibility Graph is a graph of inter visible locations, which included the start point, goal point, and vertices of obstacles(Li et al., 2002). All the points are connected with straight lines. If the straight line does not pass through the obstacle, then the connection can be said to be the Visibility Line and then call the resulting connection graph as a Visibility Graph(Kulbiej,2018). If the ship is regarded as a point, the Visibility Points can be composed of the vertices of the obstacle polygons, but the actual ship has a certain size. For the safety of the ship, it is necessary to expand the vertices of the obstacle polygons by a certain distance. The Visibility Points expansion algorithm is the same as formula (1). As shown in Fig. 4, only the Visibility Points of the convex points of the polygon are useful for route planning, and the Visibility Points of the concave points can be removed. The number of Visibility Points can be reduced by nearly a half, which will reduce the search time for the Visibility Graph. The formula (4) can be used to determine whether the vertex of the polygon is a convex point or a concave point.

Fig. 3 depicts a process diagram of Visibility Graph modelling based on electronic charts. First, the obstacle polygons are extracted from the electronic chart. The expanded polygons are created from these polygons. These expanded polygons will be used to check whether the expanded vertices are Visibility Point, and to check whether the line which connects each Visibility Point is a Visibility Line. Then, each vertex is checked to determine whether it is a convex. If it is a convex point, the expanded point will be calculated by the formula (1). If the expanded point is outside all expanded polygons, the expanded point is added to the set of Visibility Points. Finally, each Visibility Point will be connected by a straight line. If the straight line does not pass through the expanded obstacle, then the straight line will be added to the set of Visibility Lines, and the Visibility Lines connection graph is a Visibility Graph.

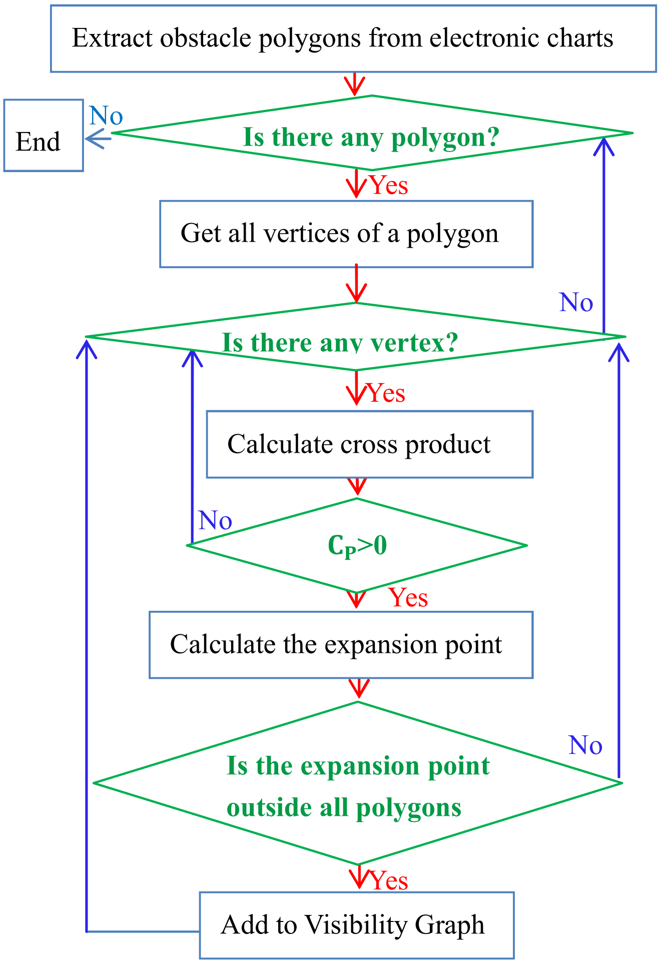


Fig. 3 Flowchart of Visibility Graph modelling based on electronic chart

In Fig. 3, The C_P is the cross product of the vector formed by the vertex and its front and back vertices. Its calculation formulas can be referred to formula in Eq. (5).

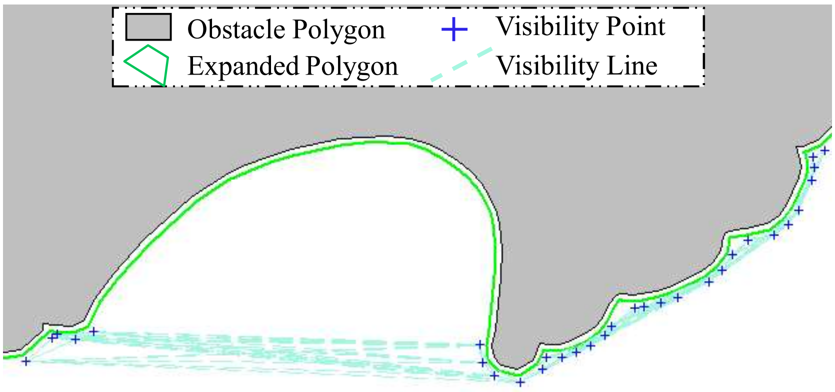


Fig. 4 Visibility Graph modelling

After Visibility Graph modelling for the electronic chart shown in Fig. 1 and connecting all Visibility Lines, the Visibility Graph is obtained, as shown in Fig. 4. The blue crosses are the Visibility Point of the electronic chart, and the cyan dashed line indicates the Visibility Line.

3. Modelling of multi-scale charts

The storage, definition, description, and maintenance of two or more scale maps of geographic elements in a map database is the so-called multi-scale expression of geographic elements. Multi-scale maps use different resolutions to express the geometric, topological and attribute information of the same element at different levels of detail so that users can choose one or more maps of different resolutions according to specific needs (Wang and Li, 2014). In order to convert multi-scale charts into Visibility Graph models, it is necessary to partition, simplify, organize, and manage the multi-scale charts so that computer programs can automatically load the required chart data to prepare for subsequent ship route planning.

In this paper, a new method is proposed for the rapid planning of long-distance routes based on global charts. In order to improve the speed of planning and the accuracy of planning routes, three different scales of chart data are used, namely small-scale map, medium-scale maps, and large-scale maps. The scale of small-scale map is 1: 110,000,000, or 1 cm = 1,100 km. The scale of medium-scale maps is 1: 50,000,000, or 1 cm = 500 km. The scale of large-scale maps is 1: 10,000,000, or 1 cm = 100 km. In this Section, the small-scale map will be simplified, which will reduce the number of Visibility Points. The medium large-scale maps will be partitioned, which will increase the speed of Visibility Graph modelling.

3.1. Simplification of small-scale map

The time-consuming function of route planning based on Visibility Graph can be described by O (n^2) (Kaluđer, 2011), where n represents the number of Visibility Points. When the number of Visibility Points n increases, the route planning time increases by n^2. Therefore, in order to shorten the route planning time, it is necessary to reduce the number of Visibility Points n as much as possible. This section will discuss how to simplify a small-scale map and establish its Visibility Graph model.

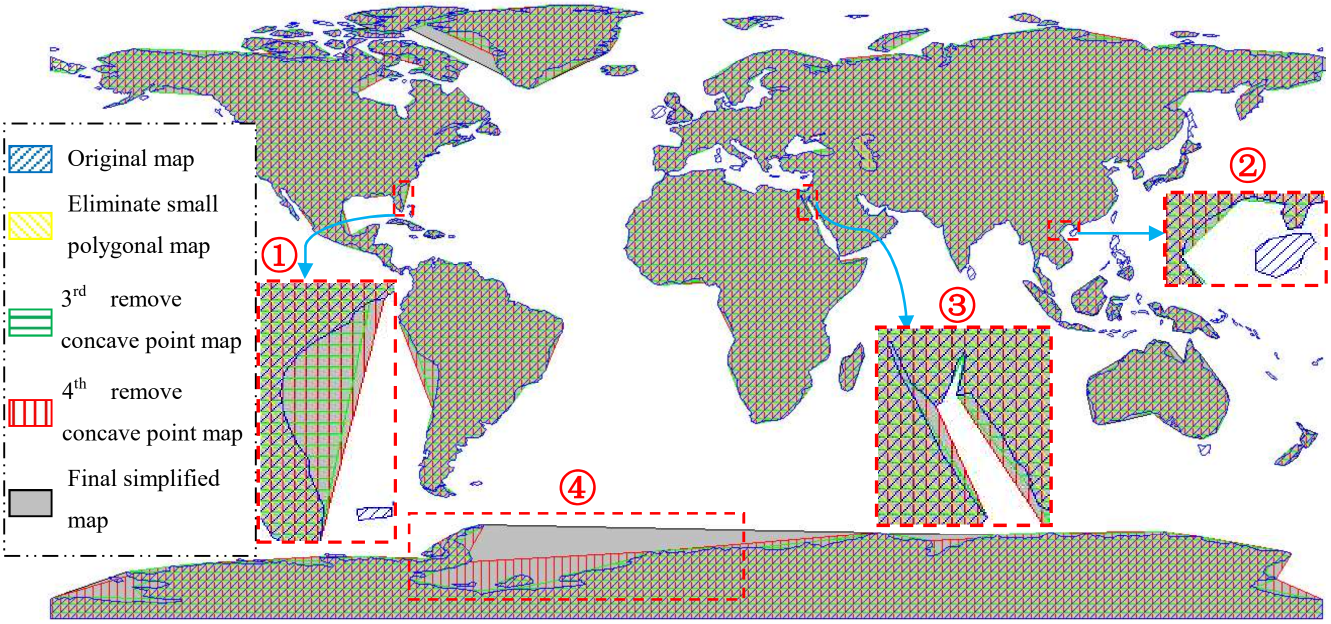
The principle of map simplification is as follows. On the basis of preserving the shape of the original map as much as possible, the vertices of the each polygons will be deleted if they have little influence on the route planning. Based on this principle, this paper proposes to remove the concave points that change the shape of the polygon less than convex points do. The so-called "small change in shape" of the polygon is determined by calculating the size of the triangle area formed by the concave point and the two points before and after it. If the triangle area is less than a given threshold, the concave point is deleted in the polygon. If the concave point is at a high latitude, the deletion threshold of its triangular area can be increased. In addition, if the triangle is a narrow and long triangle (that is, the minimum angle of the three angles is smaller than the given threshold angle, it is judged as a narrow and long triangle), the concave point can also be deleted. Because there are many small islands in the map, the number of Visibility Points is large. This paper proposes to remove the polygons where the longitude range of the island is less than 4.5 degrees and the latitude range is less than 2.25 degrees.

The calculation time of the computer will vary with different CPU configuration changes. The computer configuration used in this paper is as follows: The processor is Intel (R) Core (TM) i5-4300M CPU @ 2.60GHz, and the memory (RAM) is 8.00GB. The system type is a 64-bit operating system. Here, the small polygons, the concave points of "small change in shape", and the cross-edge of polygons are simplified by using small-scale map data. The simplified data is shown in Table 1:

Table 1. Simplified results of small-scale charts

Simplified method	Number of polygons	Number of vertices	Number of Visibility Points	Time-consuming of modelling [s]
small-scale map	128	5143	2915	38.95
Eliminate small polygonal map	38	4253	2308	23.20
Remove concave points map	36	2300	1510	9.63
Final simplified map	32	1918	1312	7.50
Reduction rate compared to the original small-scale map	75%	62.7%	55%	80.7%

As shown in Table 1, by removing small-area polygons and removing concave points, the number of polygons can be greatly reduced. The total number of vertices on the map and the number of Visibility Points are reduced after modelling. Compared to the original small-scale map, the final simplified map reduced the number of polygons by three quarters, the number of vertices by 62.7%, the number of visible points by 55%, and saved 80.7% of planning modelling time. The simplification of the small-scale map greatly improved the efficiency of ship route planning. The overlay of the simplified process map is shown in Figure 5:



Original map
Eliminate small polygonal map
3rd remove concave point map
4th remove concave point map
Final simplified map
①
②
③
④

Fig.5 Overlay of the simplified small-scale map

As shown in Figure 5, according to this simplified algorithm, most of the outline of the world map has not changed, only the polygons with local concave points have changed. It can be seen from the comparison of ①and ②that when the concave shape of the polygon is changed slowly, the concave point will be simplified. When the concave point is in a narrow and long triangle of the polygon, the point is simplified by comparison ②with ③. It can be seen from the comparison of ③and ④that when the concave points of the polygon are at high latitudes, the concave points with a large triangular area will also be simplified.

3.2. Partition and management of medium-scale map

Medium-scale maps are not simplified, therefore their details of the original details are retained. However, due to the more detailed terrain when compared to small-scale maps, the number of polygon vertices is larger. In order to improve the modelling speed and reduce the time of route planning. A discussion on how to partition and manage of the medium-scale map will be given. This paper proposes to divide the medium-scale map data into 4 sub-maps and manage the data of these sub-maps using 4 folders. The latitude and longitude data of the map are used to partition the map data. On the 0-longitude line, the map is cut from 90 degrees north latitude to 90 degrees south latitude, and cut on the equator line (that is, 0-latitude line), thus the map is divided into four parts. However, the two dividing lines are not completely straight lines, and vary in a local area according to the terrain. The purpose of doing this is to minimize changes to the original map. If the two dividing lines are do not zigzag, more polygons will be divided. The segmentation of the medium-scale map is shown in Figure 6:

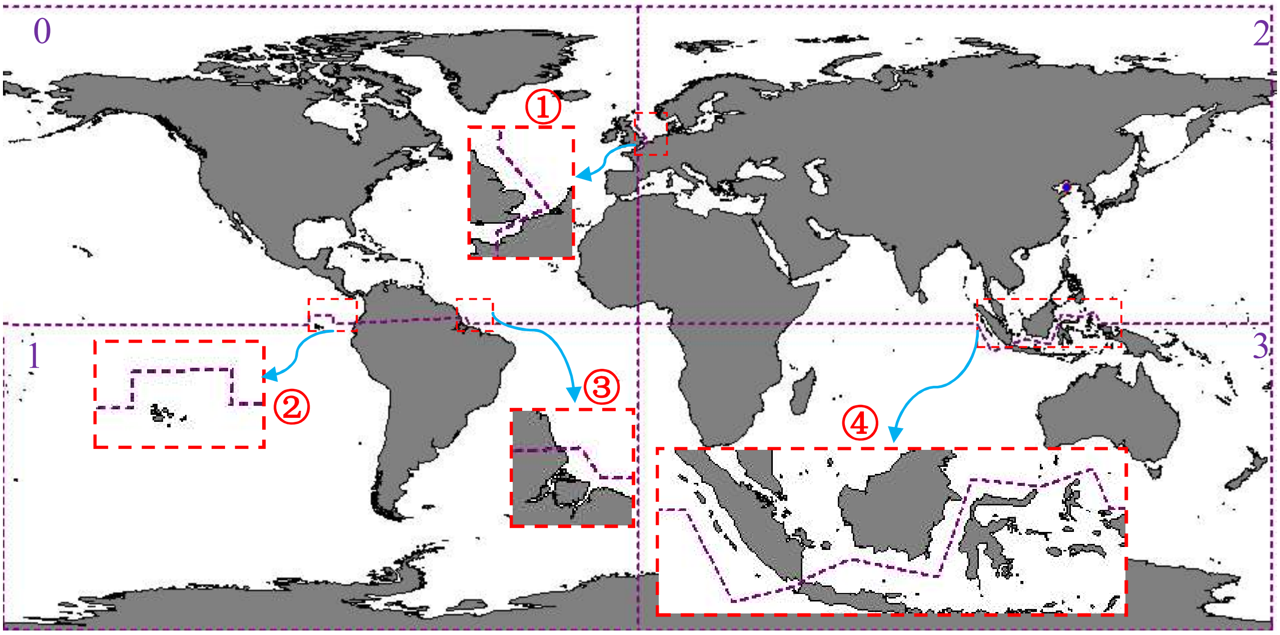


Fig.6 Partition of medium-scale map

There are four polygons shown using the dash purple lines in Fig. 6. The four polygons are formed by the two zigzagging lines and the four boundary lines of the medium-scale map. The medium-scale map is cut into four sub-maps by these four polygons, and named medium-scale sub-map 0, sub-map 1, sub-map 2, and sub-map 3, respectively. The Visibility Graphs of these medium-scale sub-maps are modeled by using these four sub-maps. The modelling parameters of these sub-maps are compared with the whole modelling of the medium-scale map as shown in Table 2:

Table 2 Comparison of modelling parameters for medium-scale maps

Map name	Number of polygons	Number of vertices	Number of Visibility Points	Computational time of modelling [s]
Sub-map 0	438	24676	14786	64.52
Sub-map 1	201	6970	4228	5.65
Sub-map 2	521	19712	11356	44.85
Sub-map 3	271	9091	5417	10.33
Sum of 4 sub-maps	1431	60449	35787	**125.35**
Medium-scale map	1427	60441	35779	**381.92**
Reduction rate compared to the original medium-scale map	-	-	-	**67.19%**

As shown in Table 2, as the number of polygon vertices increases, the computational time of Visibility Points modelling of the chart increases, because each Visibility Point needs to be compared with all polygons to determine whether the vertex is not inside all polygons. The sum of the computational time of modelling the four sub-maps is 125.35 seconds, while the modelling of the whole medium-scale map takes 381.92 seconds on the computer used. It can be seen that separate modelling of the medium-scale maps can result in modelling savings of 67.19% in terms of the time.

3.3. Partition and management of large-scale map

In the same way as the partition of medium-scale map, this paper does not simplify maps for large-scale map, and completely retains the details of the original map. On the basis of the division of the original medium-scale map, each medium-scale sub-map is divided into four sub-maps. Therefore, the large-scale map data will be divided into 16 sub-maps and named large-scale sub-map 0 to sub-map 15. Then the data of these sub-maps is managed in 16 folders separately. The present approach follows the principle of "minimizing changes to the original map". These dividing lines are not completely straight, but are tortuous in local areas according to the terrain. The dividing lines and partition of the large-scale map are shown in Figure 7:

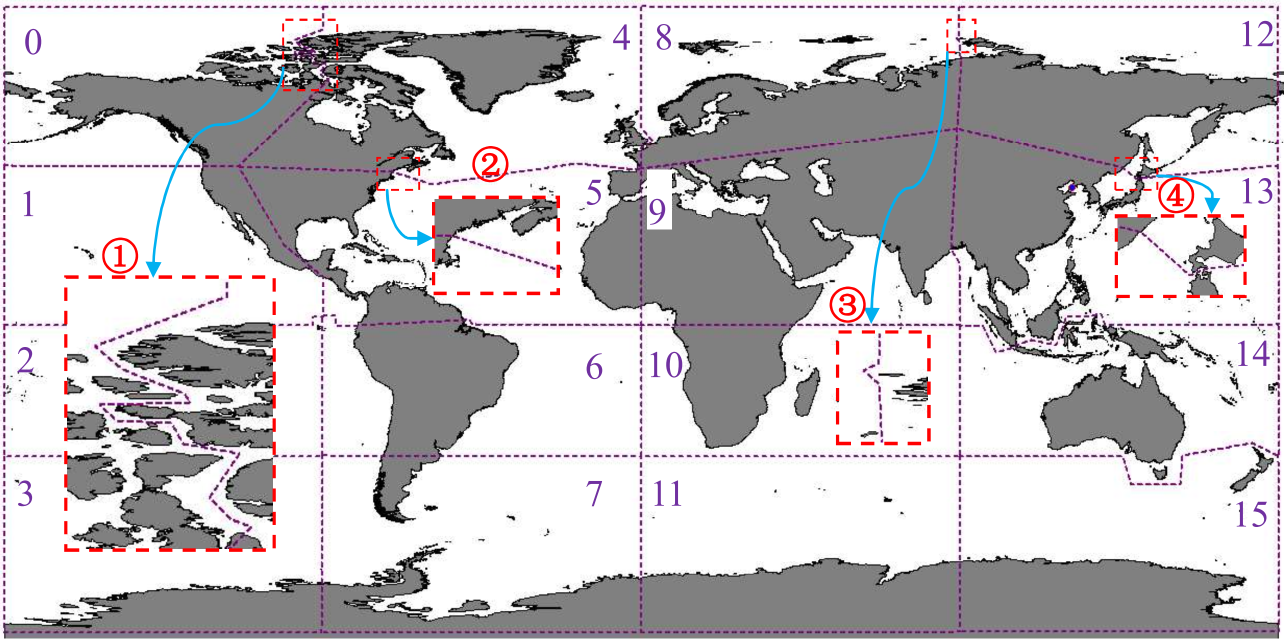


Fig.7 Partition of large-scale map

As shown in Fig.7, sixteen polygons are composed of the zigzag dividing lines and four boundary lines of the world map. The sixteen polygons are shown by the purple dotted line in Fig. 7. At this point, the small-scale map, medium-scale map, and large-scale map of the global map are all modeled, and the route planning method based on the multi-scale Visibility Graph model will be discussed in next part.

4. Route planning method based on multi-scale Visibility Graph model

In the previous sections, the Visibility Graph models and data file management of three maps with different scales were implemented. Each data folder contains three files: the obstacle polygon data file, the expanded obstacle polygon data file, and the Visibility Points data file. Only the expanded polygon data file and Visibility Points data file are used in route planning. From Table 2, it can be seen that there are thousands of Visibility Points for each chart. If many Visibility Points are used to search for the best route from, the computational time of searching is excessive. However, among the many Visibility Points, only a few Visibility Points around the route are useful for the route search. Therefore, this article proposes a method of Local Planning Window. Only those Visibility Points near the route are selected for the Visibility Graph. The method of Local Planning Window will be discussed below.

4.1. Method of Local Planning Window

The planning window is moved to the route segment, according to the needs of the route planning, and the Visibility Points inside the window are used to search for the optimal route. This reduces the number of the Visibility Points in Visibility Graph model, and shortens the search time, as shown in Fig.9. The construction of the Local Planning Window requires the following three processes:

(1) Obtain the start point and goal point of the route plan.

The start point $\mathrm{P_S}$ and goal point $\mathrm{P_G}$ will be obtained from the desired route. The size of the Local Planning Window is determined by the position of start point and goal point. $\mathrm{D_{Lon}}$ represents the distance of the currently planned route in the longitude direction. And $\mathrm{D_{Lat}}$ represents its distance in the latitude direction. These parameters can be calculated using the following formulae.

$$\begin{cases} D_{Lon} = \mathrm{Max}(P_S.x, P_G.x) - \mathrm{Min}(P_S.x, P_G.x) \\ D_{Lat} = \mathrm{Max}(P_S.y, P_G.y) - \mathrm{Min}(P_S.y, P_G.y) \end{cases} \tag{6}$$

In which, the Max() function is to take the maximum value of two numbers, and the Min() function is to take the minimum value of two numbers.

(2) Create the Local Planning Window.

The position and size of the Local Planning Window rectangle R_{LPW} are determined according to the start point and goal point of the current route plan. The calculation formula is as follows:

$$\begin{cases} R_{LPW}.Top = \mathrm{Max}(P_S.y, P_G.y) + C_E * D_{Lat} \\ R_{LPW}.Left = \mathrm{Min}(P_S.x, P_G.x) - C_E * D_{Lon} \\ R_{LPW}.Right = \mathrm{Max}(P_S.x, P_G.x) + C_E * D_{Lon} \\ R_{LPW}.Bottom = \mathrm{Min}(P_S.y, P_G.y) - C_E * D_{Lat} \end{cases} \tag{7}$$

In which, $\mathrm{C_E}$ represents the expansion coefficient of the Local Planning Window. The larger the coefficient, the larger the Local Planning Window. The number of Visibility Points will be changed by changing the value of expansion coefficient.

(3) Select Visibility Points add to Visibility Graph.

The position of each Visibility Point in the current map data will be compared with the newly generated Local Planning Window, and it will be determined whether the Visibility Point is in the Local Planning Window, and if so, the Visibility Point will be added to the Visibility Graph. In addition, the start point and goal point of the route plan should be added to the Visibility Graph.

(4) If route planning fails, increase the expansion coefficient.

Sometimes, the collision-free path cannot be searched, because the range of obstacle crossed by the route is larger than range of the current Local Planning Window. Therefore, it is necessary to increase the size of the Local Planning Window. From Equation 7, it can be seen that if the value of C_E is increased, the size of the Local Planning Window can be increased. More Visibility Point will be added to the Visibility Graph by increasing the coefficient until the planning is successful.

In order to illustrate the essence of the Local Planning Window, a simple example is used for the port chart data shown in Fig. 1. The route planning does not use the method of Local Planning Window shown in Fig. 8, and the method is used in Fig. 9.

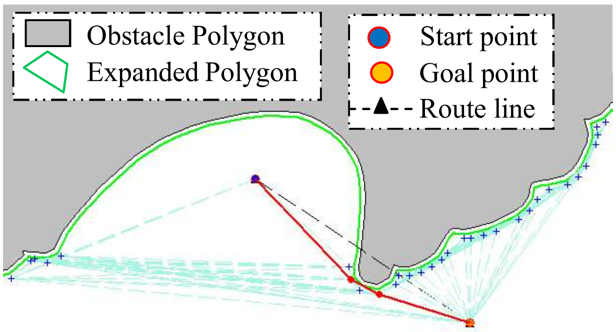


Fig.8 Route planning using general method of VG

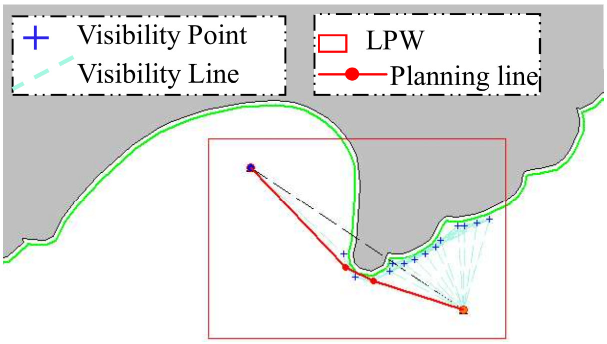


Fig.9 Route planning using the method of LPW

As shown in Fig. 8, all Visibility Points are added to the Visibility Graph. There are one obstacle polygon and 33 Visibility Points used in the route planning. The path searching time is 0.637s. In Fig. 9, the method of Local Planning Window is used. Only the Visibility Points located in the Local Planning Window are added to the Visibility Graph. The number of Visibility Points is 17, which takes 0.189s CPU time to compute. In this example, the route planning time is reduced by 70.33% by using the method of Local Planning Window.

4.2. Decompose the longer route with the great circle route method

It can be seen from the method of Local Planning Window that if the distance between the start point and the goal point is farther, the Local Planning Window is also larger, and more Visibility Points will be added to Visibility Graph. In this case, the purpose of reducing planning time cannot be achieved. Therefore, this article will use the great circle route method to decompose the longer route segment into shorter route segments, and then use the method of Local Planning Window. Here we will discuss how to use the great circle route method to decompose a longer route segment.

The great circle route is the shortest course between two points on the surface of the earth. This is most useful in long-distance navigation as it will save time and fuel(Roh,2013). However, due to the limitation of the actual control of the ship, if you follow the great circle route, the direction angle must frequently change, making steering very inconvenient. Therefore, the method of approaching the great circle route is usually used. Generally, the great circle route is divided into several sections, then the middle waypoint is obtained, and a constant direction line is used to navigate from one waypoint to the next waypoint. If the start point P_S and goal point P_G are known, and the circle needs to be equally divided into N segments, the process of calculating a great circle route is as follows:

(1) Calculate the great circle voyage.

Using the start point P_S and goal point P_G, the great circle voyage S can be obtained using the spherical triangle cosine theorem:

$$S = \arccos[\sin\ (P_S.y)\sin\ (P_G.y) + \ \cos\ (P_S.y)\cos\ (P_G.y)\cos\ (P_G.x - P_S.x)\ ] \qquad (8)$$

In which, $P_S.x$ and $P_S.y$ are the longitude and latitude of the start point respectively. When the formula(8) is used, the latitude and longitude should use the radians, and S is the radian of the great circle voyage.

(2) Calculate the initial course of the great circle route.

The initial course of the great circle route is solved based on the start point P_S, which can be obtained by the sine theorem of the spherical triangle, as follows :

$$C = \arctan[\frac{\sin\ (P_G.x - P_S.x)}{\cos\ (P_S.y)\tan\ (P_G.y) - \sin\ (P_S.y)\cos\ (P_G.x - P_S.x)}] \tag{9}$$

The initial course C of the great circle route shall be treated as follows:

$$C = \begin{cases} C & C \geq 0 \\ \pi + C & C < 0 \end{cases} \tag{10}$$

(3) Calculate the middle waypoint.

The total voyage S will be divided into M segments according to a certain distance. The voyage of each segment is d = S / M. Based on the start point P_S, the voyage of the N-th intermediate waypoint P_N is $S_N = Nd$. According to the starting course C of the great circle route, the calculation formula of the geographic coordinates of the intermediate waypoint is as follows:

$$\begin{cases} P_N.y = \ \arcsin(\sin(P_S.y)\ \cos(S_N) + \cos(P_S.y)\ \sin(S_N)\ \cos(C) \\ P_N.x = P_S.x + \arccos[\frac{\cos(S_N) - \sin(P_S.y)\ \sin(S_N.y)}{\cos(P_S.y)\ \cos(S_N.y)}] \\ \end{cases} \tag{11}$$

4.3. Meteorological optimization

When sailing in the ocean, the ship will encounter various dangerous phenomena, such as large roll angle, large acceleration, surf-riding and broaching-to. This will result in a loss of speed. In severe cases, it may lead to capsizing or causing damage to ship hull, cargo(Cai, 2014). Therefore, it is necessary to plan routes to avoid areas with severe sea conditions and reduce sailing time.

The influence of wind and waves on ship speed is difficult to determine, but statistical methods can be used to identify its influence coefficients from a large amount of navigation data. In low-speed winds (less than 20 kn), ships lose speed in headwinds and increase speed slightly in tailwinds. For higher speed winds, the speed of the ship will decrease in both headwinds and tailwinds. This is due to the increase in wave action. Wave action is the cause of ship motion. This motion reduces propeller thrust and increases resistance due to steering correction. In (Panigrahj, 2012; Tsou, 2013) the formula of speed loss was used to optimize route in wave. Although the impact of wave on ships is much greater than that of wind, it is difficult to separate the two during ship navigation. (Liu, 1992; Chu, 2008) not only considers the influence of waves on ship speed, but also considers the influence of wind on ship speed. The formula is as follows:

$$V = V_0 - \left(a_1 h - a_2 q h + a_3 F \cos\alpha\right)\left(1.0 - a_4 D V_0\right) \tag{12}$$

In which, V is the actual speed(kn) of ship in the sea, which is determined by the performance of ship and the sea conditions. V_0 is the speed in calm water. h is the wave height(m). q is the angle(°) between ship heading and wave direction. F is wind speed(m/s). α is the angle(°) between ship heading and wind direction. a_i is the coefficient determined by the ship's performance. D is the actual displacement(t) of the ship.

4.4. Search algorithm

There are many search algorithms for shortest paths, such as Dijkstra, A* algorithm, fuzzy logic method, topological method, penalty function method, genetic algorithm, simulated annealing method, ant colony method, neural network method, etc. This article uses one of the classic algorithms of the shortest path algorithm, the Dijkstra algorithm, which gets a wide attention as it solves an important problem of graph theory (Khan, 2016). This algorithm is suitable for calculating the shortest path problem with non-negative distance weights. Its advantage is that the search ideas are clear and the result is accurate.

Firstly, it will be determined whether the connected line between two Visibility Points intersects with any polygons. If any polygon is intersected, the corresponding line is an un-visibility line. If there is no intersection, this line is Visibility Line. The voyage time matrix is shown in Figure 10, then the Dijkstra algorithms is used to find the path of shortest voyage time.

$$\begin{array}{r c}
 & \begin{array}{cccccc} Start & p_1 & p_2 & p_3 & p_j & Goal \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \end{array} \\
\begin{array}{r} Start \rightarrow \\ p_1 \rightarrow \\ p_2 \rightarrow \\ p_3 \rightarrow \\ p_i \rightarrow \\ Goal \rightarrow \end{array} &
\begin{bmatrix} 0 & T_{01} & T_{02} & T_{03} & \cdots & T_{0n} \\ T_{10} & 0 & T_{12} & T_{13} & \cdots & T_{1n} \\ T_{20} & T_{21} & 0 & T_{23} & \cdots & \vdots \\ T_{30} & T_{31} & T_{32} & 0 & \cdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & T_{ij} & \vdots \\ T_{n0} & T_{n1} & \cdots & \cdots & \cdots & 0 \end{bmatrix}
\end{array}$$

Fig.10 Voyage time matrix for search algorithm

As shown in Fig. 10, "Start" is the start point, " Goal" is the goal point, "n" is the number of the Visibility Points. The point $\mathrm{P_i}$ and point $\mathrm{P_j}$ are the Visibility Points. "T_{ij}" is the voyage time between point P_i and point P_j. The formula of distance T_{ij} is shown as follows:

$$T_{ij} = \begin{cases} 2R_E \arcsin(\sigma) / V_{ij} & \textit{Visibility Line} \\ T_{Max} & \textit{Un-visibility Line} \end{cases} \tag{13}$$

In which, $R_E \approx 6371\text{km}$ is the value of the mean earth radius. T_{Max} is a very big voyage time, it is mean that the connected line between two Visibility Points is a un-visibility line. V_{ij} is the voyage speed from point P_i to point P_j. It can be calculated by the formula (12). And σ can be calculated as follows:

$$\begin{cases} \sigma = \sqrt{\sin^2(\frac{\Delta y}{2}) + \cos(\mathrm{P_i.y}) \cos(\mathrm{P_j.y}) \sin^2(\frac{\Delta x}{2})} \\ \Delta x = P_j.x + \mathrm{P_i.x} \\ \Delta y = P_j.y + \mathrm{P_i.y} \end{cases} \tag{14}$$

In which, $P_i.x$ and $P_i.y$ are the geographical longitude and latitude in radians of point P_i. And $P_j.x$ and $P_j.y$ are the geographical longitude and latitude in radians of point P_j.

4.5. Route planning process based on multi-scale maps

In order to realize the long-distance and rapid route planning for autonomous ships, it will be discussed how to organize these multi-scale Visibility Graph models in an orderly way, and step by step to achieve a long-distance, fast search of the optimal route. As shown in Fig. 11, this paper adopts gradual route planning. Firstly, the small-scale map is used to plan a rough route; Then the medium-scale maps are used to modify the route; Finally, large-scale maps are used to refine the route.

In the process of small-scale map planning, the latitude and longitude of the start and goal points are obtained, and the size and position of the Local Planning Window are calculated. Then it is determined whether the Panama

Canal and Suez Canal are in the Local Planning Window. If so, the artificial waypoints of the canal are added to Visibility Points, establishing the Visibility Graph model of the small-scale map. Finally, the shortest path will be searched in the Visibility Graph matrix. In the planning process of medium-scale and large-scale maps, the medium-scale or large-scale global chart model are established, then the route is planned using the established Visibility Graph model. Due to the different scales of the maps, the locally planned routes need to be optimized. The optimized route is decomposed by using the great circle route method. Because the great circle route planning does not consider the map model, the route must be checked and planned again. Finally, the shortest route is renewed. The internal planning process of medium-scale and large-scale maps is shown in Figure 12:

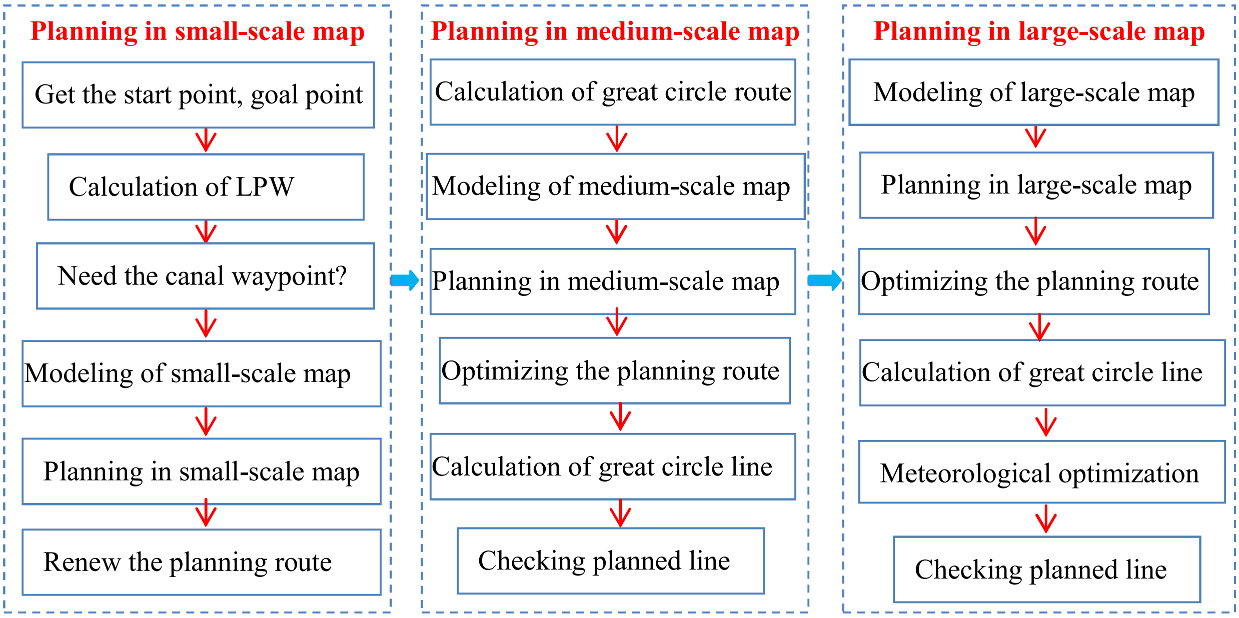

Planning in small-scale map
Get the start point, goal point
Calculation of LPW
Need the canal waypoint?
Modeling of small-scale map
Planning in small-scale map
Renew the planning route
Planning in medium-scale map
Calculation of great circle route
Modeling of medium-scale map
Planning in medium-scale map
Optimizing the planning route
Calculation of great circle line
Checking planned line
Planning in large-scale map
Modeling of large-scale map
Planning in large-scale map
Optimizing the planning route
Calculation of great circle line
Meteorological optimization
Checking planned line

Fig. 11 Flow chart of route planning based on multi-scale Visibility Graph method

On the basis of planning a rough route on a small-scale map, it is now necessary to divide the route into a small section to plan section by section. If the section of the route is too long, it will get a larger Local Planning Window. The larger the Local Planning Window, the greater the number of Visibility Points, which will lead to time-consuming path planning. The longer route segment is decomposed into smaller route segments, using the great circle route method. At this time, it is assumed that there are M segments of routes. Figure 12 describes the planning process of the Nth segment route in medium-scale and large-scale map.

As shown in Fig.12, firstly, the latitude and longitude of the start point, goal point from the Nth route are obtained. Since the medium-scale and large-scale maps have multiple sub-maps, it is necessary to determine which sub-map the Nth route is located in. After obtaining the sub-map number where the Nth route is located, all the polygon data and Visibility Points data of the sub-map are loaded. Before calculating the Local Planning Window, it needs to determined whether the current start point and goal point are inside the polygons. If they are, the points are replaced by a point with the nearest Visibility Points to avoid the failure of the route planning of this section. Then the size and position of the Local Planning Window are calculated, by which its Visibility Graph model is established. Following this, the shortest path is searched in the Visibility Graph model, using Dijkstra algorithms. If the shortest path length is greater than the given maximum distance D_{Max}, there is no accessible path, and the size of the Local Planning Window needs to be enlarged until the shortest accessible path is obtained. This is then repeated for planning the next route, until all route segments are planned.

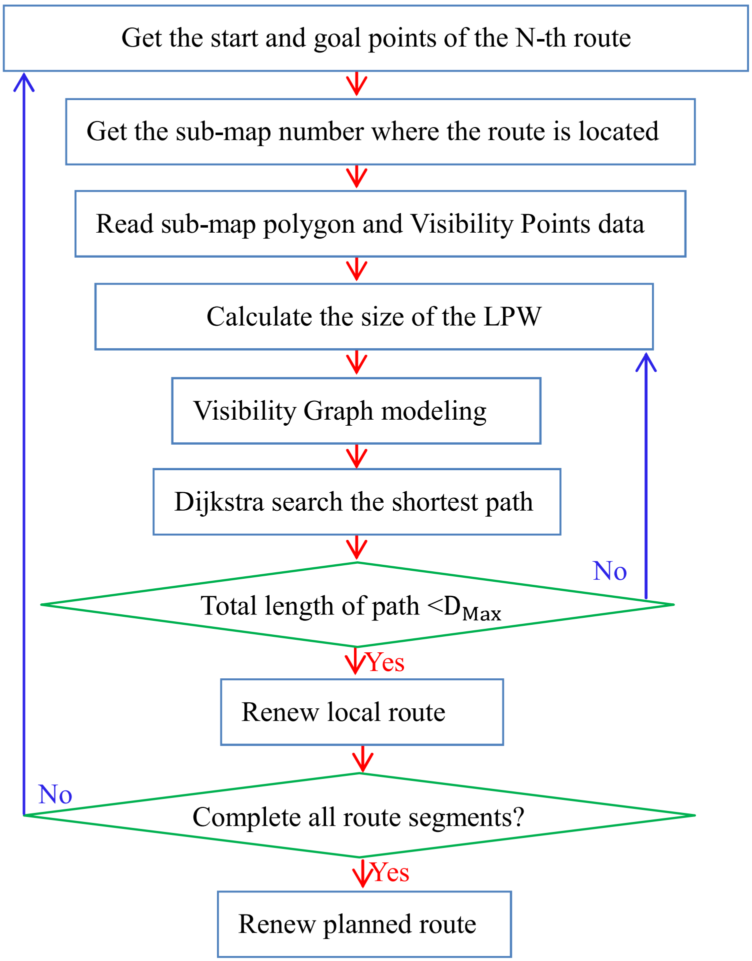

Get the start and goal points of the N-th route
Get the sub-map number where the route is located
Read sub-map polygon and Visibility Points data
Calculate the size of the LPW
Visibility Graph modeling
Dijkstra search the shortest path
No
Total length of path <D_Max
Yes
Renew local route
No
Complete all route segments?
Yes
Renew planned route

Fig.12 Local route planning in medium-scale and large-scale map

5. Long-distance route planning for Intelligent ship

In order to illustrate the feasibility of the route planning method based on the multi-scale Visibility Graph method proposed here, the route from Dalian Port in China to the Port of Rotterdam in Netherlands will be planned. In this paper, the ship "LongLin" was taken as an example, which parameters are given in Table 3, starting from Dalian Port, passing through Singapore, passing through the Suez Canal, entering the Mediterranean Sea, and finally reaching the port of Rotterdam. Before long-distance route planning, the global chart data needs to be preprocessed. First, the global land and island geographic data and shallow water geographic data with a water depth of not more than 10 meters need to be obtained. In this paper, the expansion of the polygons of land and islands is used to approximate the geographic data of shallow seas.

Table3 The primary parameters of "LongLin" ship

Length overall	153.6(m)	Displacement	13,433(t)
Beam	21.83 (m)	Engine speed	120(RPM)
Draft	9.42(m)	Speed in calm water	20(kn)

Liu (1992) had selected 105 sets of actual ship observation data from the ship's logbook, including heading, still water speed, wind direction, wind speed, wave direction, wave level, actual ship speed and deadweight. The least square method was used to identify the wind and wave influence coefficients a_i in the formula (12), were $a_1 = 1.08, a_2 = 0.126, a_3 = 2.77 \times 10^{-3}, a_4 = 2.33 \times 10^{-7}$. These parameters were used to calculate the voyage time in formula (13) in this paper.

5.1. Route planning in small-scale map

The effective establishment of the map model is a prerequisite for effective route planning. The modelling of small-scale maps is based on the scale of 1: 110,000,000, and the simplification process is used to reduce the Visibility Points of the global map to about 1200, which will ensure fast planning the long-distance route. When the map is simplified, the polygons of obstacles will be expanded to get the expanded polygons. Here, all polygons are expanded outward by 5000 meters, but the Visibility Points are expanded outward by 10,000 meters based on the expanded polygon. The small-scale map is modeled using these parameters. Dalian Port is select as the start point and Rotterdam Port as the goal point for small-scale map planning. The planning result is shown in Fig.13:

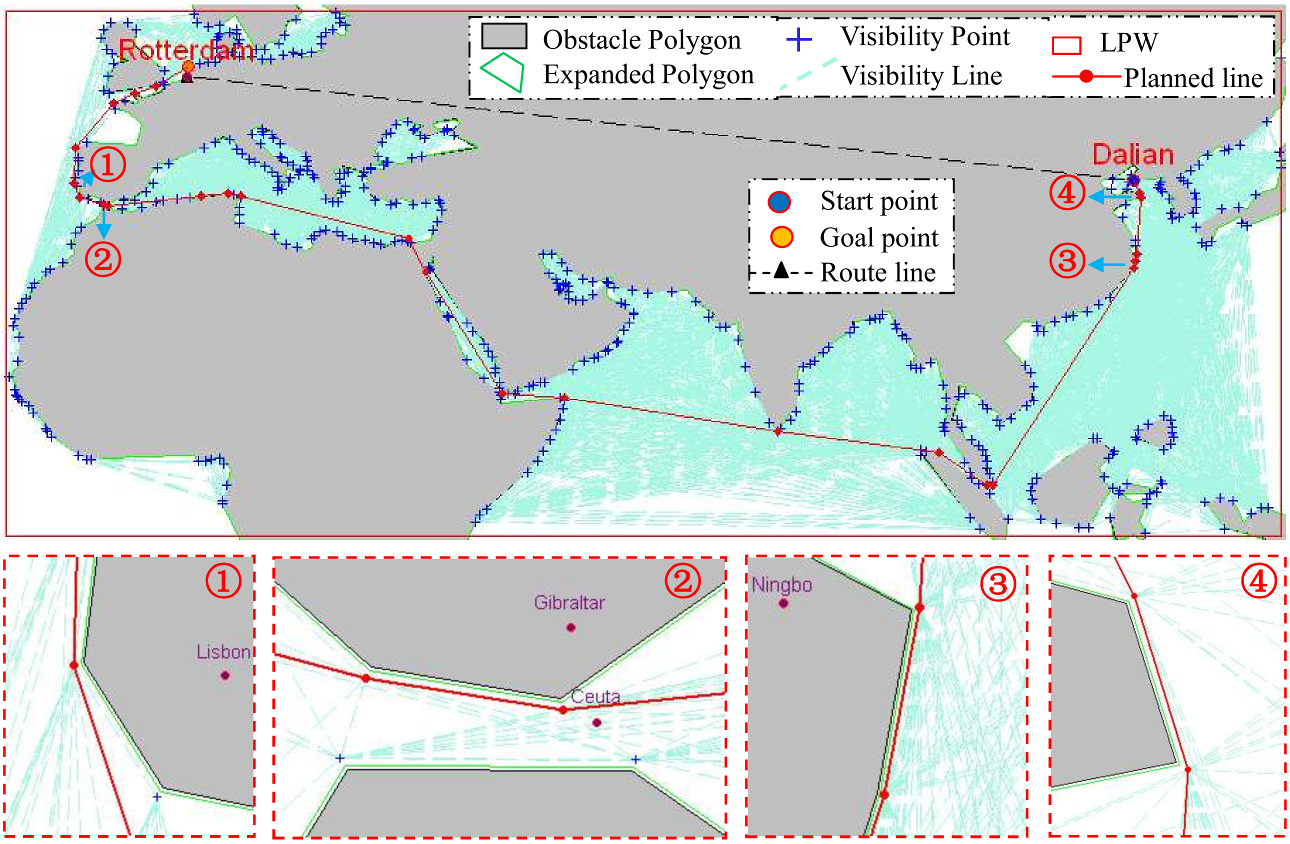


Fig.13 Route planning in small-scale map

The model data used in the planning of the small-scale map are shown in Table 4. The global small-scale map contains 32 polygons and 1312 Visibility Points, while the actual planning uses 9 polygons (polygon usage rate is 28.13%) and 492 Visibility Points (Visibility Points usage rate is 37.5%), the modelling and planning of small-scale maps takes 7.878 seconds. In this paper, the Local Planning Window method is used to reduce the Visibility Points by 62.5% in small-scale map planning, so the Local Planning Window method can reduce a large number of Visibility Points, thereby saving planning time. As shown in Figure 13, the red circled numbers are used to mark the key locations in the plan, and these partially planned routes are shown in an enlarged view below the planning map. The enlarged view ①describes the planned route near Lisbon, the enlarged view ②shows the planned route in the Strait of Gibraltar, the enlarged view ③shows the planned situation near Ningbo, the enlarged view ④shows the planned route near the Shandong Peninsula in China. All these enlarged views will be compared with the planning route in the medium-scale maps and large-scale maps. The route planning process based on the Multi-scale Visibility Graph method can now be described.

5.2. Route planning in medium-scale map

Firstly, the medium-scale map is based on the 1: 50,000,000 scale, and it is partitioned. In this paper, the medium-scale map data is divided into four sub-maps, and the data of these sub-maps are managed in four folders. Then, the polygons of the obstacles are expanded. Here, all polygons are expanded outward by 1000 meters, and the Visibility Points are expanded outward by 2000 meters on the basis of the expanded polygons. According to these parameters, the medium-scale map is modeled. Finally, based on the small-scale planned route, the secondary planning route using the medium-scale map model is carried out. The model data used in the medium-scale map planning are shown in Table 4, and the planned route is shown in Fig. 14:

Table 4 The planning result using small-scale and medium-scale map

Index	Planning process	Number of polygons	Number of Visibility Points	Local planning times	Computational time of modelling or planning[s]	**Reduction rate of Visibility Points**
1	Modelling of small-scale map	32	1312	1	0.031	**-**
2	Planning in small-scale map	9	492	1	7.847	**62.5%**
3	Calculation of great circle line	-	-	25	0.001	**-**
4	Modelling of medium-scale sub-map 2	521	11356	1	0.499	**-**
5	Planning in medium-scale sub-map 2	28	284	13	1.184	**97.5%**
6	Modelling of medium-scale sub-map 0	438	14786	1	0.437	**-**
7	Planning in medium-scale sub-map 0	3	62	2	0.14	**99.58%**
8	Optimization of route line	959	-	5	0.578	**-**
9	Calculation of great circle line	-	-	49	0.312	**-**
10	Checking planned line in medium-scale map	1	12	1	0.063	**-**
	Sum the computational time of planning	**-**	**-**	**-**	**11.092**	**-**

As shown in Table 4, two of the four medium-scale sub-maps are used in this planning (the sub-maps usage rate is 50%). That is the 2nd and 0th sub-map used in this route planning. There is 13-time local planning in the 2nd sub-map. 28 polygons out of 521 polygons (polygon usage rate is 5.37%), and 284 out of 11356 Visibility Points in the 2nd sub-map are used in the 13-time local planning (Visibility Points usage rate is 2.5%). There are 438 polygons and 14786 Visibility Points in the 0th sub-map. Only 3 polygons (polygon usage rate is 0.68%) and 42 Visibility Points (Visibility Points usage rate is 0.42%) are used in the 2-time local planning process of the 0th sub-maps. The total time for modelling and planning in the medium-scale map is 3.714 seconds. Through data analysis, it can be seen that the medium-scale map is divided into 4 sub-maps, and this planning only uses 2 sub-maps, which reduces the modelling time by 50%. The number of Visibility Points are reduced by 97.5% by using the Local Planning Window method. And it also reduced the number of Visibility Points by 99.58% in the 0th sub-map, so the Local Planning Window method can reduce a large number of Visibility Points. It also shortens the route planning time.

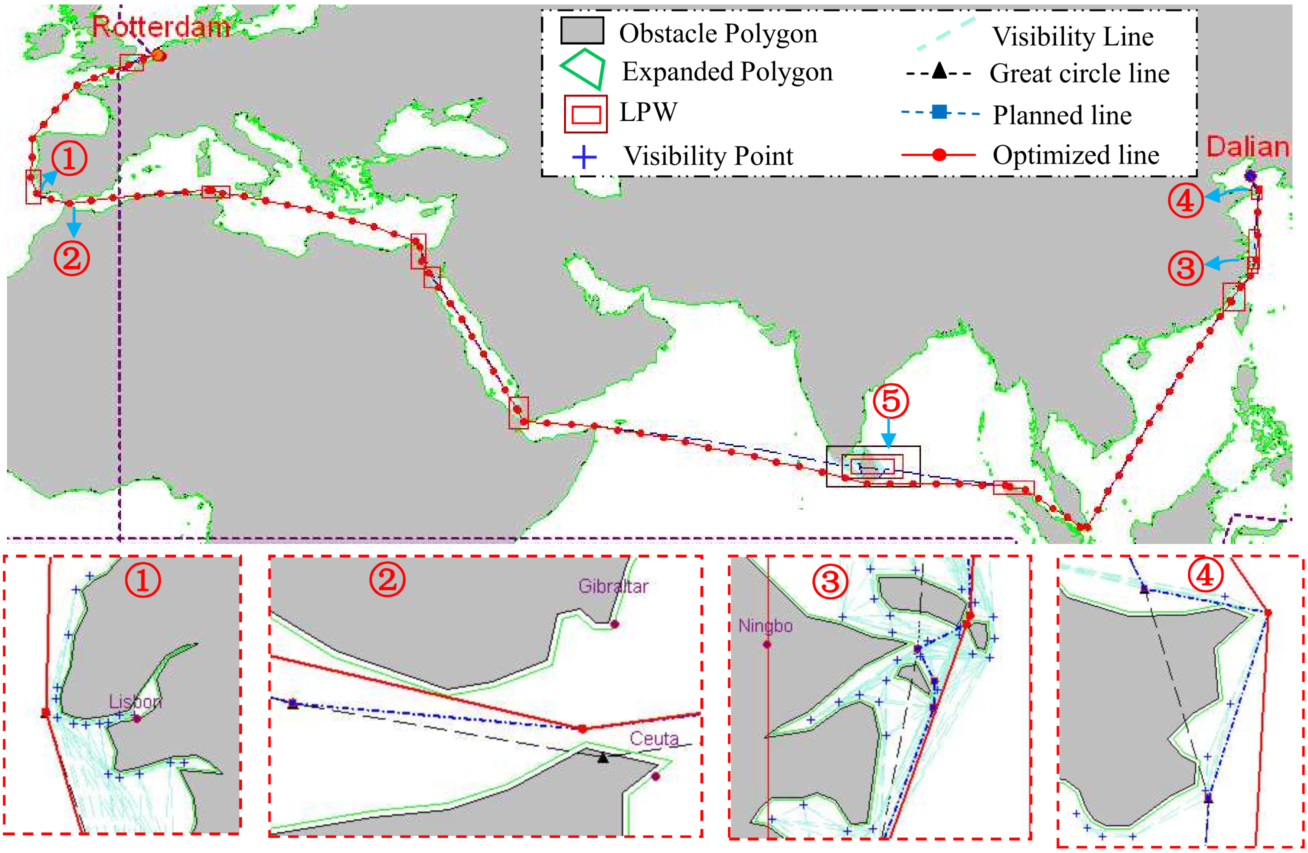


Fig.14 Route planning in medium-scale map

As shown in Figure 14, comparing the zoomed-in maps ①, ②, ③ and ④with the zoomed-in maps in the small-scale map, the medium-scale map has more detail terrain. The enlarged view ①describes the newly planned route near Lisbon. In the enlarged view ②, there is a planning point (small black triangle) of a large circle route inside the polygon, which was replaced by the nearest Visibility Points (small red circle) to avoid the failure of the route planning. In the enlarged view ③, some islands have been added to the sea area map near Ningbo. The picture shows the newly planned route. By comparing the planned route (blue square dotted line) with the optimized route (red circle solid line), it is obvious that the optimized route is shorter. The enlarged image ④ shows the updated planned route near the Shandong Peninsula in China. From this, we can clearly understand the optimization process from the great circle route, planning route, to the optimized route.

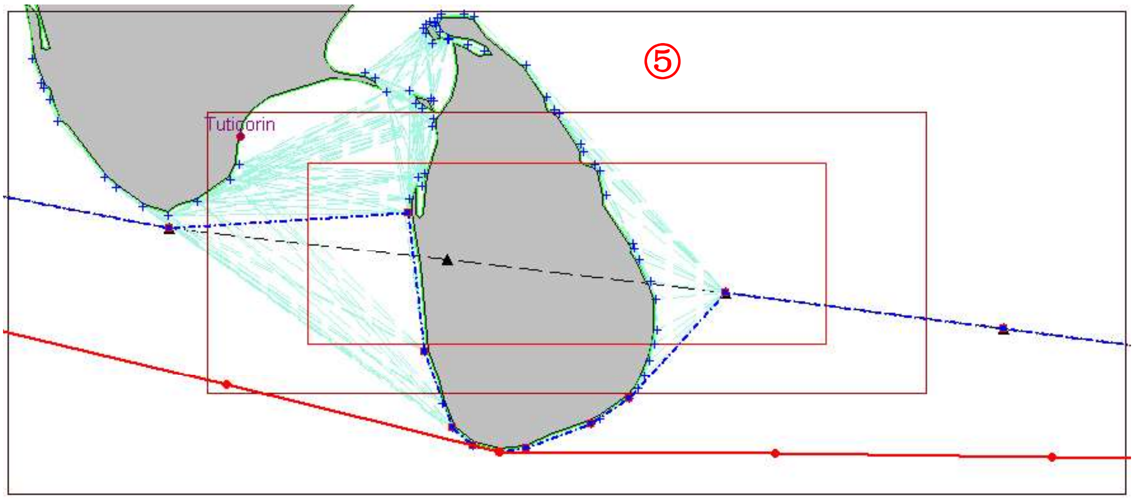
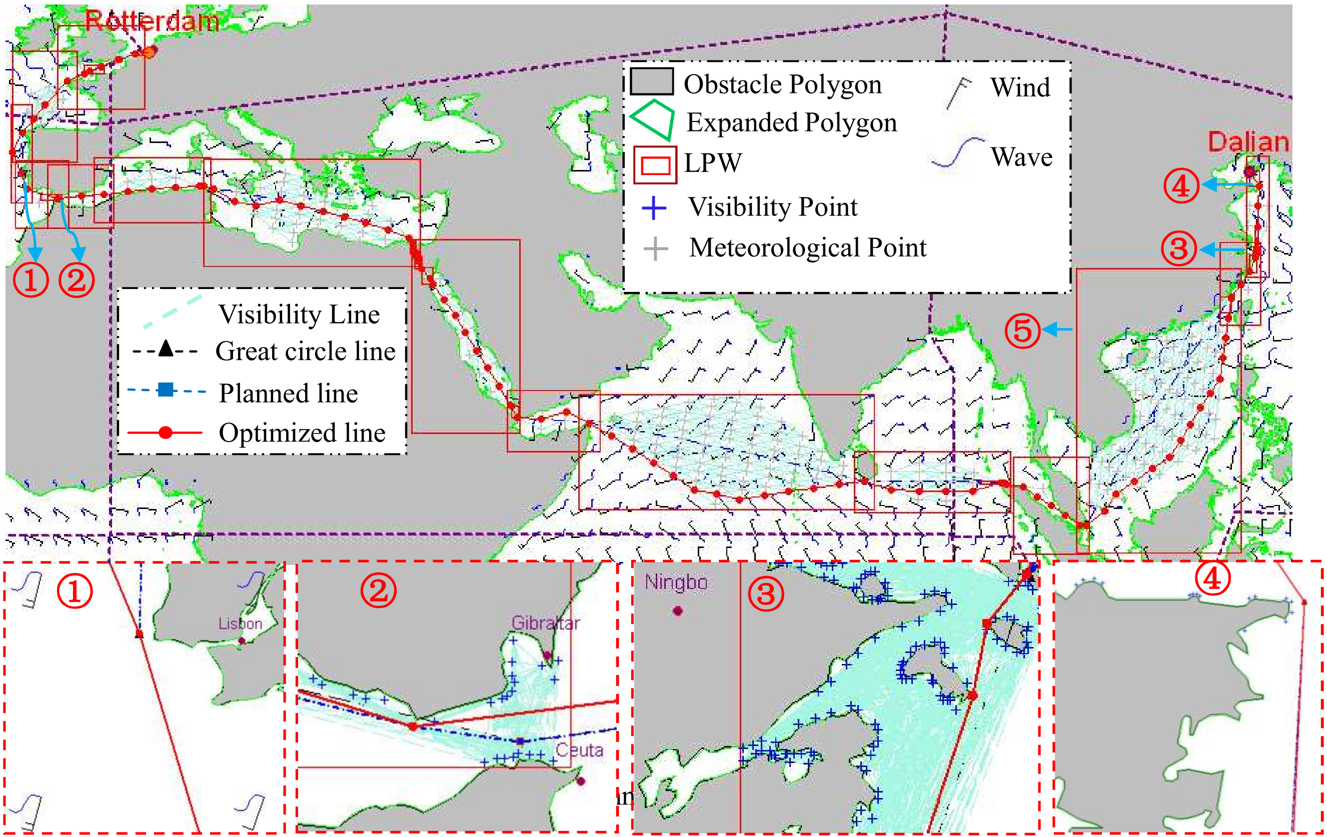


Fig. 15 An enlarged view of multi-times local planning near the Sri Lanka Island

Fig. 15 is the enlarged view ⑤in Figure 14, there are three nested Local Planning Windows. The colors of rectangular window from small to large are red, brown, and dark brown. They represent the first, second, and third times of the local planning windows. Comparing the size of the Sri Lanka island with the scope of the first and second planning windows, it is clear that there are no accessible routes for the first two times planning, until the Local Planning Window is expanded to the third one. In the enlarged picture ⑤ the changes of planned route can be clearly seen from the great circle route, the planned route, and to the optimized route.

5.3. Route planning in large-scale map

The modelling of large-scale map is based on the scale of 1: 10,000,000, and the sub-maps are divided. This large-scale map data is partitioned into 16 sub-maps, and the data of these sub-maps are managed respectively in 16 folders. After the map is divided, the obstacle polygon is expanded. Here, all polygons are expanded outward by 200 meters, and the Visibility Points are expanded outward by 400 meters on the basis of the expanded polygons. Then, on the basis of the medium-scale planned route, the 3rd time route planning of the large-scale map is carried out, and the planning result is shown in Figure 16:



Comparing the four enlarged views in Fig. 16 and Fig. 15, the terrain in the large-scale map has more details, as shown in Figure 16. It can be seen from the enlarged view ①and ④that these two areas have not re-planned routes. In the enlarged view ②, the great circle route intersects the obstacle polygon, so the local route needs to be re-planned. In the enlarged view ③, the map of the sea area near Ningbo has added some detailed islands, so the area has re-planned routes. A series of red waypoints can be seen on the Suez Canal. This article simply adds the existing Suez Canal waypoints to the route.

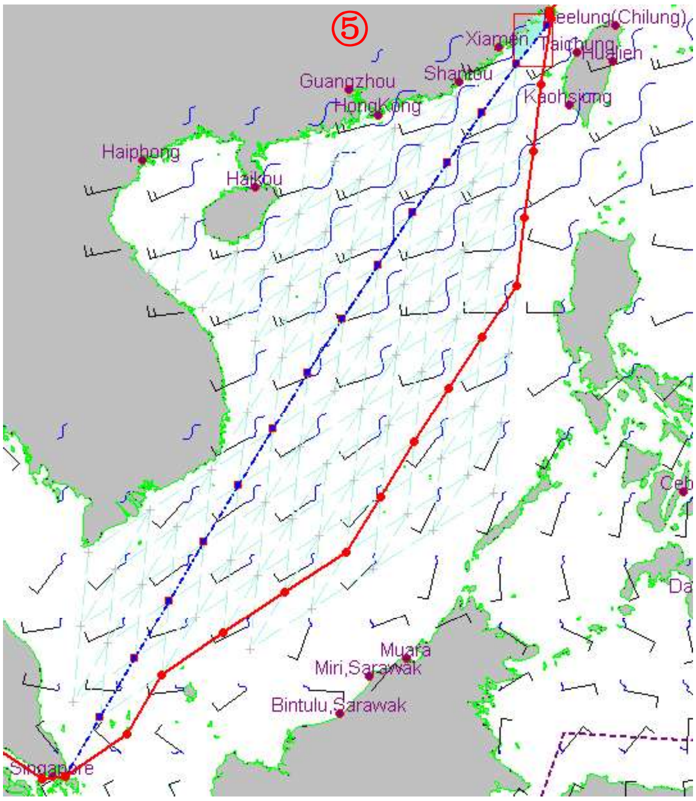


Fig.17 An enlarged view of meteorological optimization route near the South China Sea

Fig. 17 is the enlarged view ⑤ in Figure 16. The meteorological points(the light gray crosses) is generated based on the great circle route. The great-circle route is equally divided into multiple routes, and the meteorological lines are perpendicular to the great-circle route. The meteorological points are got from these meteorological lines by equally divided points. The pre-defined meteorological grid system should cover the entire area that the ship may need to sail into. It can be seen from Fig. 17 that the optimized route(red circle solid line) chooses to pass through areas with less wind and waves. This will avoid harsh sea conditions and shorten the sailing time.

Table 5 The planning result using large-scale map

Index	Planning process	Number of polygons	Number of Visibility Points	Local planning times	Computational time of modelling or planning [s]	Reduction rate of Visibility Points
1	Modelling of large-scale sub-map 13	680	25571	1	1.28	-
2	Planning in large-scale sub-map 13	14	254	8	1.132	**99.01%**
3	Modelling of large-scale sub-map 9	409	17292	1	2.012	-
4	Planning in large-scale sub-map 9	5	61	4	0.609	**99.65%**
5	Modelling of large-scale sub-map 5	412	17489	1	1.123	-
6	Planning in large-scale	1	35	1	0.478	**99.8%**

	sub-map 5					
7	Modelling of large-scale sub-map 4	460	41295	1	2.59	-
8	Planning in large-scale sub-map 4	1	22	1	0.125	**99.95%**
9	Modelling of large-scale sub-map 8	370	20369	1	1.669	-
10	Planning in large-scale sub-map 8	2	31	1	0.437	**99.85%**
11	Optimization of route line	2331	-	3	1.763	-
12	Calculation of great circle line	-	-	64	1.606	-
13	Meteorological optimization	270	299	13	2.872	-
14	Checking planned line in large-scale map	7	90	7	1.172	-
15	Adding canal line	-	-	1	0.01	-
	Sum the computational time in large-scale map	-	-	-	**18.878**	-

As shown in Table 5, in the 13th sub-map, there are 8-times local planning, and used 14 of the 680 polygons (polygon usage rate 2.06%), and 254 of 25571 Visibility Points (Visibility Points usage rate 0.99%) in the 13th sub-map. The ninth sub-map contains 409 polygons and 17,292 Visibility Points. In the local planning process, only 5 polygons (polygon usage rate 1.22%) and 61 Visibility Points (Visibility Points usage rate 0.35%) are used. Through the data analysis, it can be seen that this local planning uses 5 of the 16 large-scale sub-maps (the sub-map usage rate is 31.25%). Because the method of Local Planning Window is used, the number of Visibility Points are decreased by 99.01% in the 13th sub-map. And the number of Visibility Points are decreased by 99.65% in the 9th sub-map. The local planning in the 5th, 4th, and 8th sub-map, the number of Visibility Points decreased by 99.8%, 99.95%, 99.85% respectively. Therefore, the Local Planning Window method can reduce the number of Visibility Points, which will save the planning time-consuming. The computational time for modelling and planning in large-scale maps is 18.878 seconds. Analyzing the planning data of Table4 and Table5, the total time-consuming of route planning is 29. 97 seconds. The results show that the route planning method based on Multi-scale Visibility Graph can greatly reduce the planning time and can be applied to long-distance route planning.

6. Conclusions

This paper proposes a long-distance route planning method based on the Multi-scale Visibility Graph method. The information of the electronic chart layers is analyzed, the layer related to the route planning is selected, and those maps data are converted into polygons. Then the Visibility Graph model is established by expanding these polygons and extracting their Visibility Points. The modelling method of multi-scale maps is introduced in detail. The small-scale map is simplified by the method of removing small-area polygons and removing polygonal concave points, which will reduce the complexity of the small-scale Visibility Graph model. The medium-scale map data is divided into 4 sub-maps, and these sub-maps are managed with in 4 folders. The large-scale map is divided into 16 sub-maps, and these sub-map data are managed separately, which facilitates the computer program to automatically read the corresponding model data. In order to reduce the number of Visibility Points in Visibility Graph, this paper proposes a method of Local Planning Window. The principle is: only those Visibility Points near the route are selected to Visibility Graph. The size of the mobile planning window is determined by the distance between the two waypoints, so the method of the great circle route is used to decompose the long route segment. Then, the route planning method based on the Multi-scale Visibility Graph method is proposed. First, the small-scale Visibility Graph model is used to plan a rough route, and then the medium-scale and large-scale Visibility Graph models are used to refine the route, in which the great circle route planning, route optimization and route inspection planning are used repeatedly. Finally, the route planning from Dalian Port to Rotterdam Port is carried out. And the number of polygons and Visibility Points used in multi-scale planning are compared with the number of polygons and Visibility Points in the map. The results showed that the use of Multi-scale Visibility Graph method reduced the computational time of modelling. The use of the Local Planning Window method greatly reduced the number of Visibility Points, which greatly reduced the route planning time. Therefore, the Multi-scale Visibility Graph method can be applied to long-distance ship route planning. In the future, for different ship types, corresponding route planners will be developed. This will involve developing a database to store optimized parameters for different ship types.

Acknowledgements

This work was performed within the Department of Naval Architecture, Ocean and Marine Engineering, University of Strathclyde. The study is supported by the National Natural Science Foundation (NNSF) of China (No. 51309148), Science & Technology Program of Shanghai Maritime University (No. ZZshhs12055).

References

Bijlsma, S. J. , 2008. Minimal Time Route Computation for Ships with Pre-Specified Voyage Fuel Consumption. Journal of Navigation, 61(4):723-733.

Cai, Y., Wen Y.Q., 2014. Ship Route Design for Avoiding Heavy Weather and Sea Conditions. International Journal on Marine Navigation and Safety of Sea Transportation, 8(4):551-556.

Candeloro, M., Lekkas, A., Sørensen, A., 2017. A Voronoi-diagram-based dynamic path-planning system for underactuated marine vessels. Control Engineering Practice. Control Engineering Practice, 61, 41-54.

Chu, S.,2008. On the Speed Loss Problem when Vessels Sailing Gale and Seas. Journal of Zhejiang International Maritime College, 4(2):1-6.

Emad ,G.R., Khabir, M., Shahbakhsh, M., 2020.Shipping 4.0 and Training Seafarers for the Future Autonomous and Unmanned Ships. 21th Marine Industries Conference, January 2020, Qeshm Island.

Felski, A., Zwolak, K., 2020. The Ocean-Going Autonomous Ship-Challenges and Threats. J. Mar. Sci. Eng. 8, 41.

Gao M., Shi G., Li W., Wang Y., Liu D., 2017. An Improved Genetic Algorithm for Island Route Planning. Procedia Engineering. 174, 433-441.

Hagiwara H., 1987. Practical Weather Routing of Sail-assisted Motor Vessels. The Journal of Navigation, 40(1):96-119.

Hanssen, G. L., James, R.W., 1960. Optimum Ship Routing. Journal of Navigation, 1960, 13(3):253-272.

He, Y., Zhang, D., Zhang, J., Zhang, M.Y., Li, T.W., 2019. Ship Route Planning Using Historical Trajectories Derived from AIS Data. Journal on Marine Navigation and Safety of Sea Transportation. 13(1):69-76.

Kaluđer, H., Brezak, M., Petrović I., 2011. A visibility graph based method for path planning in dynamic environments. 2011 Proceedings of the 34th International Convention MIPRO, Opatija, pp. 717-721.

Khan, Z.A., 2016. Comparison of Dijkstra's Algorithm with other proposed algorithms (Master's thesis). Virtual University of Pakistan, Federal Government University.

Kosmas, O.T., Vlachos, D.S, Simos, T.E., 2008. Obstacle Bypassing in Optimal Ship Routing Using Simulated Annealing. International Electronic Conference on Computer, 1060(1):79-82.

Kulbiej, E., 2018. Autonomous Vessels' Path finding Using Visibility Graph.// 2018 Baltic Geodetic Congress, 107-111.

Lee, M.-C., Nieh, C.-Y., Kuo, H.-C., Huang, J.-C., 2019. An Automatic Collision Avoidance And Route Generating Algorithm For Ships Based On Field Model. Journal of Marine Science and Technology, 27(2):101-113.

Lee, S.-M., Roh, M.-I., Kim, K.-S., Jung, H., Park, J.J., 2018. Method for a simultaneous determination of the path and the speed for ship route planning problems. Ocean Engineering, 1571:301-312.

Lehtola, V., Montewka, J., Goerlandt, F., Guinness, R., Lensu, M., 2019. Finding safe and efficient shipping routes in ice-covered waters: A framework and a model. Cold Regions Science and Technology, 165, 102795.

Li, L., Ye, T., Tan, M., Chen, X.-j., 2002. Preset State And Future Development Of Mobile Robot Technology Research. ROBOT, 24(5):475-480.

Li, Y., Chen., H., 1997. Design of optimum ship route using weather routing techniques. Journal of South China University of Technology (Natural Science), 25(12):65-69.

Lin, Y.-H., Huang, L.-C., Chen, S.-Y., Yu, C.-M., 2018. The optimal route planning for inspection task of autonomous underwater vehicle composed of MOPSO-based dynamic routing algorithm in currents. APPL OCEAN RES, 75, 178-192.

Lin, Y.-H., 2018. The Simulation of East-Bound Transoceanic Voyages According to Ocean-Current Sailing Based on Particle Swarm Optimization in the Weather Routing System. Marine Structures, 59, 219-236.

Lin, Y.-H., Wang, S.-M., Huang, L.-C., Fang, M.-C., 2017. Applying the Stereo-Vision Detection Technique to the Development of Underwater Inspection Task with PSO-Based Dynamic Routing Algorithm for Autonomous Underwater Vehicles. Ocean Engineering, 139, 127-139.

Liu, F., 1992. Study on the Ship's Loss-speed in Wind and Waves. Journal of Dalian Marine College, 18(4): 347-351.

Montes, A.A., 2005. Network Shortest Path Application for Optimum Track Ship Routing (Master's thesis).

Operations Research Department Naval Postgraduate School.

Moravec, H., Elfes, A., 1985. High resolution maps from wide angle sonar. Proceedings. 1985 IEEE International Conference on Robotics and Automation, St. Louis, MO, USA, pp. 116-121.

Niu, H. Lu, Y., Savvaris, A., Tsourdos, A., 2016. Efficient Path Planning Algorithms for Unmanned Surface Vehicle, 10th IFAC Conference on Control Applications in Marine Systems, At Trondheim, Norway. September 2016.

Panigrahi, J.K., Padhy, Panigrahi, J.K., Padhy, C.P., Sen, D., Swain, J., Larsen, O., (2012). Optimal ship tracking on a navigation route between two ports: a hydrodynamics approach. Journal of marine science and technology, 17(1), 59 - 67.

Roh, M.-I., 2013. Determination of an economical shipping route considering the effects of sea state for lower fuel consumption. Naval Archit. Ocean Eng. 5:246~262.

Sen, D., Padhy, C.P., 2015. An approach for development of a ship routing algorithm for application in the North Indian Ocean region. Applied Ocean Research, 50:173-191.

Shah, B.C., Gupta, S.K., 2016. Speeding Up A* Search on Visibility Graphs Defined over Quadtrees to Enable Long Distance Path Planning for Unmanned Surface Vehicles. Proceedings of the Twenty-Sixth International Conference on Automated Planning and Scheduling (ICAPS 2016) June 2016, pp 527–535.

Silveira P., Teixeira A.P., Soares, C.G., 2019. AIS Based Shipping Routes Using the Dijkstra Algorithm. the International Journal on Marine Navigation and Safety of Sea Transportation,13(3): 565-571.

Szlapczynska, J., Szlapczynski, R., 2019. Preference-based evolutionary multi-objective optimization in ship weather routing. Applied Soft Computing , 84:1-21.

Tsou，M.-C., Cheng, H.-C., 2013. An Ant Colony Algorithm for efficient ship routing. Polish Maritime Research , 20(3): 28-38.

Wang, Y., Li, C., 2014. Online cascaded updating technology for multi-scale urban geographical data. Science of Surveying and Mapping, 039(010):48-52.

Wen, Y., Sui, Z., Zhou, C., Xiao, C., Chen, Q., Han, D., Zhang, Y., 2020. Automatic ship route design between two ports: A data-driven method. Applied Ocean Research, 96, 102049.

Zhang, C., Zhang, D., Zhang, M., Mao, W., 2019. Data-driven ship energy efficiency analysis and optimization model for route planning in ice-covered Arctic waters. Ocean Engineering, 18615.

Zhou, H., 2016. Research and Implementation of IACO-GA algorithm on dynamic ship route planning (Master's thesis).Wuhan University of Technology.