Ship weather routing based on modified Dijkstra algorithm

Xianming Zhu[1, a]& Hongbo Wang[1,b] , Zihao Shen[1,c] , Hongjun Lv[1,d]

[1]School of JiLin University, Changchun 130000, China

[a]zhuxm15@mails.jlu.edu.cn,[b]wang_hongbo@jlu.edu.cn,

[c]shenzh15@mails.jlu.edu.cn,[d]lvhj15@mails.jlu.edu.cn

Keywords: modified Dijkstra algorithm, weather routing, linked list.

Abstract. Ship weather routing is used to obtain a shortest time path or most economical path considering weather condition. Many approaches have been studied to solve ship-routing problem, and Dijkstra algorithm is quite practical among them. A modified Dijkstra algorithm using structure matrix and single linked list is applied to establish shortest time route in this passage. The weights are stored by structure matrix instead of adjacency matrix and single linked list is used for less time consumption. The modified algorithm owns a better computational complexity due to the improved methods, meanwhile, simulations are made to illustrate the result of the modified algorithm.

Introduction

Ship weather routing is used to obtain the shortest time route or the most economical route from departure to arrival point considering information of the weather states. Wind, waves and ocean currents have impacts on ship velocity in open ocean area. In this paper, the methods to improve the Dijkstra algorithm are emphasized, so only the effects from wave are discussed for simplicity.

There have been some algorithms to solve ship-routing problem, e.g. the modified isochrones method, the two-dimensional dynamic programming, the three-dimensional dynamic programming, the Dijkstra algorithm, and the genetic algorithm. With suitable digitization of an open sea area and with advent of computing power, Dijkstra algorithm is quite practical for the problem.

In order to serve the ship weather-routing better, many improved Dijkstra algorithms have been presented. E.W. Dijkstra (1959) [1] proposed the Dijkstra algorithm by a note on two problems in connection with graphs. Peter Hart, Nils Nilsson and Bertram Raphael (1968) [2] first described the A* (star) algorithm that it was an extension of Dijkstra algorithm. A* algorithm increased the speed of Dijkstra algorithm by using heuristics. Zhan and Noon (1996) [3] tested 15 of the 17 shortest path algorithms and identified a set of three shortest path algorithms that run fastest on real road networks. These three algorithms are: (1) TQQ (the graph growth algorithm implemented with two queues), (2) DKA (the Dijkstra algorithm implemented with approximate buckets), and (3) DKD (the Dijkstra algorithm implemented with double buckets). The TQQ algorithm is based on the theory of graph growth and use two FIFO queue to implement a deque structure to support the search process. The DKA algorithm and DKD algorithm are based on Dijkstra algorithm and use the bucket structure to improve search speed. Zheng et al. (2009) [4] proposed sector Dijkstra algorithm to limit search area and reduce the nodes to be traversed. Zhang et al. (2011) [5] presented an improved new heap algorithm called Pairing heap to optimize Dijkstra algorithm. The improved new heap algorithm solves the problem that Dijkstra algorithm takes too much time to traverse the node table, delete the minimum node and update the node table. In this paper, two dimensional structure array and single linked list are used to improve the Dijkstra algorithm.

Meteorological Data

Weather information of a specific sea area is a preliminary in routing activity. Wind, wave and current all have effects on ship velocity. R.W. James (1957) [6] found that the most important parameter retarding a ship's progress was surface wave action. Therefore, this paper considers

waves as weather condition. Wave data are downloaded from ECMWF.

Modified Dijkstra Algorithm

Ship weather routing is related to optimization of several factors, namely: minimum passage time, minimum fuel consumption, minimum structural damage and so on. These factors are related to speed reduction which includes the involuntary speed reduction and the voluntary speed reduction [7]. In this paper, the optimization object is passage time.

Dijkstra Algorithm

Dijkstra algorithm is essentially a kind of greedy algorithm. The path is defined as a series of connected vertices and the length of the path is arithmetic sum of the corresponding link lengths in the path. Consider a directed graph: $G = (V, E)$, with n vertices and e real valued weights sides. V is a collection of initial vertices. E is a collection of real valued weights sides. S is a collection of vertices which have found the shortest path from the starting node to themselves. V-S is a collection of vertices which have not found the shortest route from the starting vertex to themselves. The specific algorithm steps are as follows:

Step1. Use the weighted adjacency matrix arcs to represent the directed graph and $arcs(s, i)$ is the weight from node s to i. Suppose S equals $\{V_s\}$ and V_s is the starting node. Suppose $dist[i]$ equals the shortest distance from node V_s to V_i.

$$dist[i] = \begin{cases} 0 & i = s; \\ arcs(s, i) & i \neq s, < V_s, V_i >\in E; \\ \infty & i \neq s, < V_s, V_i >\notin E; \end{cases} \quad (1)$$

Step2. V_j is the end of the next shortest path. Select the node V_j by:
$$dist[j] = Min\{ dist[k] \mid V_k \in V - S\}, \ S = S \cup \{V_j\}.$$

Step3. If $dist[j] + arcs(j, k) < dist[k]$, then:
$$dist[k] = dist[j] + arcs(j, k), (\forall V_k \in V - S).$$

Step4. Repeat step 2 and step 3 until $S = V$.

Expression of the Weight

The sea area studied in this passage is a proportion of the North Pacific. It lies from $0°N$ to $45°N$, from $120°E$ to $180°E$ and is divided by 1*1 degrees latitude-longitude line. Each intersection of latitude-longitude line is a vertex, so there are 2806 vertexes in total. Each vertex has been assigned a number from 0 to 2805 and has eight weights except ones lie on boundaries. The optimization object in this paper is time, so the weights represent the passage time form one vertex to its direct linked vertex:
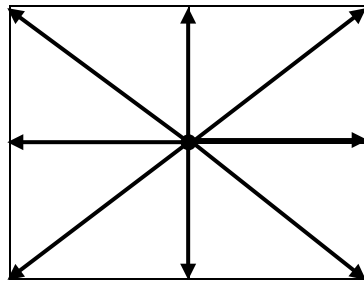

Fig. 1 Eight weights distribution of a vertex

The weights are expressed as: *w=L/V. L* is the distance between two neighboring vertex. *V* is calculated by the empirical formula which is recommended by Soviet Central Maritime Research Institute. The formula is expressed as follow [8]:
$$V = V_0 - (0.745h - 0.257qh) \times (1.0 - 1.35 \times 10^{-6}DV_0) \quad (2)$$

Where: *V* denotes the actual speed in the sea, V_0 represents the speed in the calm water, *h* means the wave height, *q* is the angle between ship heading and wave direction, *D* is the actual displacement of the ship (tons). This formula is applied to ships with displacement ranging from 5000 to 25000 tons, speeds between 9 and 20 knots, and the wave height is between 0 and 5 meters. The ship parameters used in this paper are showed in the following table:

Table 1 Ship parameters	
TYPE	tanker
LOA	196(m)
DWT	12000(tons)
SPEED	18(knots)

In original Dijkstra algorithm, the weights are stored in adjacency matrix. However, in this case, using adjacency matrix will cause a huge waste in storage. Assume N to denote the total vertexes in the sea area, then the total storage needed in adjacency matrix grows by $N*N$. Meanwhile in each line of the matrix, there will be only eight meaningful values, and the rest are wasted. To solve this, a two dimensional structure matrix is used. Each element in the matrix contains an array of eight weights, representing the weights at each vertex, also several other necessary variables are included. In this way, the total storage needed grows by $c*N$, c is a constant. So, by comparing the adjacency matrix, a large storage space can be saved.

Single Linked List

In original Dijkstra algorithm, the array *dis[i]* is used to store the total weight from the starting vertex to the corresponding vertex according to 3.1 . During the execution of the algorithm, each element in *dis[i]* has to be traversed to get the minimal weight vertex. Both storage needed and time consumption grow by $c*N$. c is a constant. For a better computational complexity, a single linked list is used to improve this.

Let *dis_list* denote the single linked list used in the algorithm. The *dis_list* is sorted during initialization. The info member of each node contains the number of the vertex and the weight from source vertex to corresponding vertex. Note that *dis_list* is sorted by weight, so the complexity of finding the minimal weight vertex is O(1), better than visiting every element in *dis[i]*. The length of *dis_list* is changeable while the algorithm is running. It can also save storage space. In this specific case, 2806 vertexes in total, the length of the *dis_list* is around 100 nodes.

Results

Two simulations in different weather conditions are presented in the following.

In figure 2, the start vertex is $A(29°N, 124°E)$, the end vertex is $B(10°N, 168°E)$. In figure 3, the start vertex is $A(25°N, 125°E)$, the end vertex is $B(5°N, 159°E)$. In the two figures, the black line represents the shortest time route. Obviously, the modified Dijkstra algorithm can find an optimal route in a certain weather condition, meanwhile, due to the improved methods, the algorithm can run at a better efficiency.
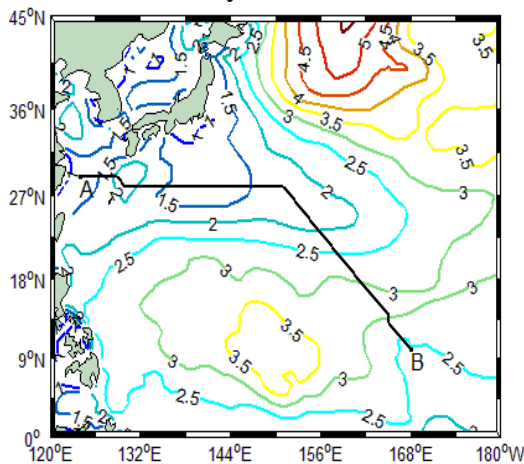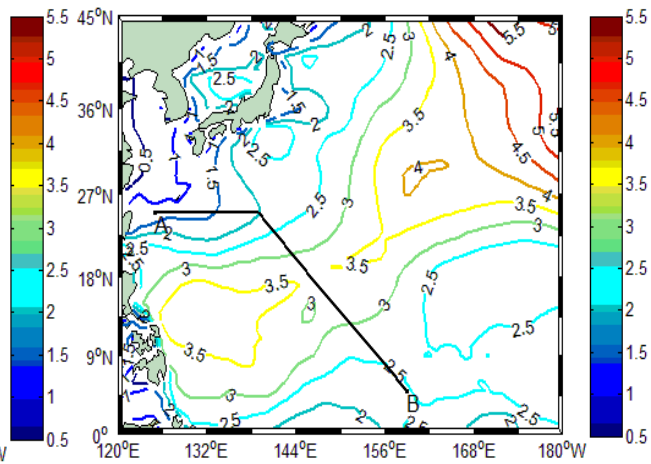

Fig. 2 simulation result　　　　　　　　　　Fig. 3 simulation result

Conclusions

It is well known that the ship routing for oceangoing vessels is widely applied, and a considerable number of studies have been carried out. In this paper, the ocean area to be navigated

is divided by latitude-longitude line, and an improved Dijkstra algorithm is proposed. The improved Dijkstra algorithm adopts structure matrix and single linked list. Storage will be saved by applying structure matrix. At the same time, single linked list will help reduce time consumption. The improvement of efficiency is obvious when there are a large number of vertexes. The simulation results of improved algorithm show that it can improve the search efficiency and find the global optimal route.

References

[1] E.W. Dijkstra. 1959. A Note on Two Problems in Connexion with Graphs. Numerische Mathematik 1, 269-271(1959).

[2] Masoud Nosrati*, Ronak Karimi, Hojat Allah Hasanvand. Investigation of the * (Star) Search Algorithms: Characteristics, Methods and Approaches. World Applied Programming, Vol. 2, No. 4, April 2012. p. 251-256.

[3] F. Benjamin Zhan. Three Fastest Shortest Path Algorithms on Real Road Networks: Data Structures and Procedures. Journal of Geographic Information and Decision Analysis, Vol. 1, No. 1, p. 69-82.

[4] S.F. Zheng, J.D. Cao, X.M. Lian. Sector Dijkstra Algorithm for Shortest Routes between Customers in Complex road networks. J Tsinghua Univ (Sci & Tech), Vol. 49, N0. 11, 2009.

[5] W. Zhang, C. Jiang. Optimization Studies on an Improved Dijkstra Algorithm. Proceedings of the 2011 3rd International Conference on Information Technology and Scientific Management (ICITSM 2011).

[6] G.J. Haltiner, H.D. Hamilton. March. Minimal-Time Ship Routing. Journal of Applied Meteorology. Vol. 1, No. 1, March 1962.

[7] J.K. Panigrahi, S.K. Misra, P.A. Umesh. Application of Oceansat-1 MSMR analysed winds to marine navigation. International Journal of Remote Sensing, Vol. 31, No. 10, 2010, p. 2623-2627.

[8] M.C. Tsou, H.C. Cheng. An Ant Colony Algorithm for Efficient Ship Routing. Polish Maritime Researsh. Vol. 20, 2013,p. 28-38.