ISSN: 2414-1895 DOI: 10.6919/ICJE.202102_7(2).0053

Unmanned Vessels Path Planning Based on A* Algorithm

Shilong Xuan

School of Navigation and Naval Architecture, Dalian Ocean University, Dalian 116023, China.

12701923@qq.com

Abstract

As the continuous development of unmanned vessels, high-level autonomous navigation in the marine environment has become the key technology of intelligent ships. As a heuristic algorithm, A* algorithm has the ability to select the optimal path in a short time. This paper mainly studies the global path planning with static marine environmental information, establishes the environmental model, finds the optimal path through A* algorithm in the vast sea area to reduce energy consumption, and finally verifies the feasibility of the method through experimental simulation.

Keywords

Unmanned Vessels; Path Planning; A* Algorithm.

1. Introduction

Path planning is the key problem of unmanned vessel[1] planning and navigation, and also being the guarantee of unmanned vessel safe and reliable navigation. Path planning[2]which is planning a feasible collision free path from the starting point to the target point marks the level of intelligent unmanned vessel and plays a key role in the speed and efficiency of maritime searching. So It has become a trend to introduce intelligent algorithm into the path planning of unmanned vessel. Simulated annealing algorithm [3] is derived from the simulation of the actual solid annealing process, with the decreasing of temperature parameters, the global optimum is found by combining the sudden jump characteristic. While the disadvantages of SA are slow convergence time and long optimization time. Genetic algorithm [4][5] uses fitness value to evaluate population individuals and cross mutation to mate. Genetic algorithm is greatly affected by the early fitness value, relatively poor solid stability, and prone to premature phenomenon or slow convergence speed and other defects. Ant colony algorithm [6] according to the idea of ants in nature, through the pheromone concentration to determine the next search path point. The disadvantage of ant colony algorithm is that it has a long search time for large-scale complex problems, and it will stagnate. In view of the shortcomings of the above algorithm, this paper introduces A*[7] algorithm to design the path generation of unmanned vessel. A* algorithm is the product of combining heuristic rules with Dijkstra algorithm. The main advantage of A* algorithm is that it can search the path quickly, plan the route better, and provide a theoretical reference for related research.

2. Problem description

2.1 Methods of environment modeling

In order to verify the effectiveness of the algorithm, it’s necessary to model according to the sea environment. This paper adopts grid method [8]. Grid method divides the moving space into several rectangular matrix regions of the same size, which are called grids. If there are obstacles in a grid range, it is an obstacle grid, otherwise it is a free grid. For each grid, a numerical value is used to

ISSN: 2414-1895
DOI: 10.6919/ICJE.202102_7(2).0053

indicate whether there are obstacles. As shown in figure 1, the white area is a free grid, and the black area is an obstacle grid.

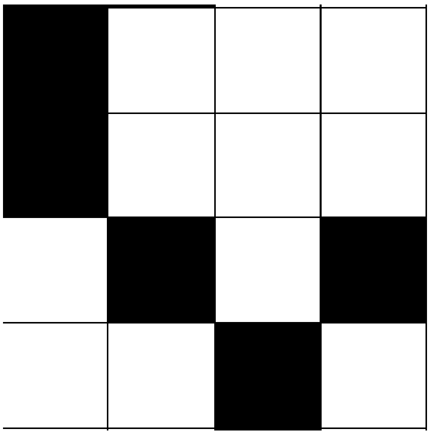

Fig.1 Grid picture

2.2 Dijkstra algorithm

Dijkstra algorithm [9] is a single source shortest path algorithm suitable for non negative weight networks. It is widely used in solving the shortest path problem. The algorithm can find the shortest distance from a point to any other vertex. The basic idea of Dijkstra algorithm is as follows:

(1) Define two tables named open and closed. Open is used to store the unexpected points and closed is used to store the inspected vertiles.

(2) Put the starting point S in the closed table and the rest in the open table.

(3) Calculate the distance between the vertices in the closed table and other fixed points, and put the shortest distance points into the closed table.

(4) A new fixed point is added to the closed table, the shortest path in the standard is updated.

(5) Back to step (3), until the empty of closed list.

As shown in figure 2, Dijkstra algorithm is used to search the shortest distance of randomly generated 10 points. 1 is the starting point and 2 is the end point.

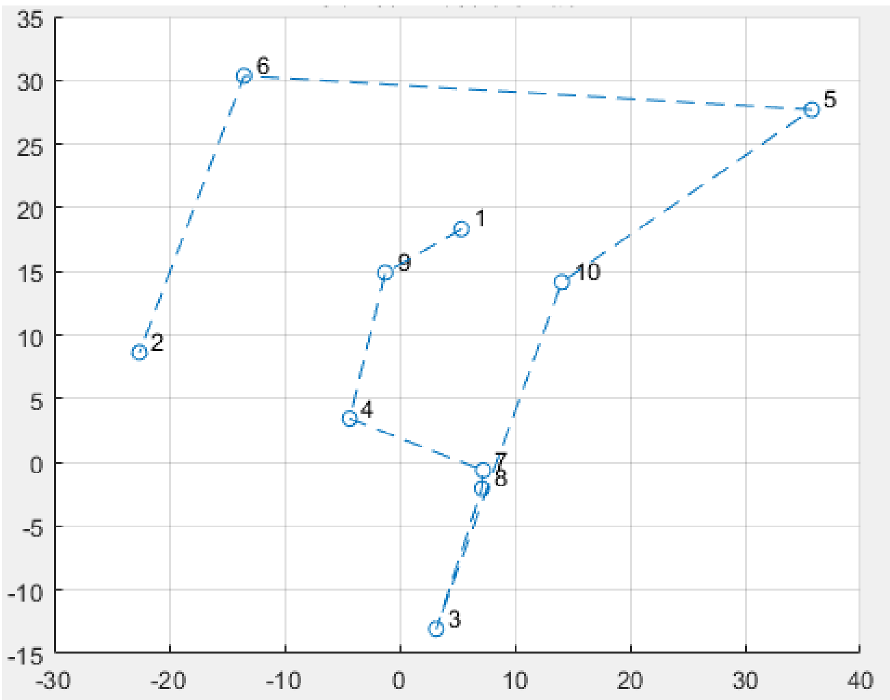

Figure.2 Path search of Dijkstra algorithm

ISSN: 2414-1895 DOI: 10.6919/ICJE.202102_7(2).0053

Dijkstra algorithm has a high time complexity and a large number of searching points because its search path is point by point. If we search in a large area of sea, the time complexity will increase geometrically. The efficiency of finding the optimal path is low. In view of the non intelligence of the algorithm, this paper introduces a heuristic algorithm based on Dijkstra algorithm to form a * algorithm to study the path planning of unmanned vessel.

3. A*algorithm

3.1 A*algorithm introduction

The A* algorithm[10] is widely used as a heuristic intelligent algorithm in the path planning of mobile robots. The idea of the algorithm is also to locate the target according to the closed linked list and the open list. The difference with the Dijkstra algorithm is the introduction of the heuristic function h (n) (Equation 1), h (n) is the estimated cost of the best path from the current node to the target node. Guided by the estimated cost, the current point will proceed in the direction of the smallest estimated value, forming an intelligent heuristic search, and finally search to the end point in a shorter time to complete the shortest path planning task.

$$\mathrm{f(n)} = g(n) + h(n) \tag{1}$$

3.2 Implementation of A * algorithm

The process of A * algorithm search path is as follows:

(1) Create open and closed tables, open stores unexamined vertices, and closed tables are used to store the examined points;

(2) Put the starting point into the open table and clear the closed table;

(3) Determine if the open table is empty, if yes, end the search as an end instruction;

(4) Determine whether the searched point is the target point, if it is, the search is successful and jump out of the loop;

(5) If it is not, the F value of the extended point of the current point is calculated to determine whether the current extended point exists in the open table or the closed table;

(6) If it is not in the open and closed tables, choose to store it in the open table;

(7) If it exists in the open table, update the F value in the table. Take the Fmin value as the parent node for the next round of search;

(8) Repeat step (5) to find the target point and jump out of the loop.

As shown in Figure 3, the A * algorithm performs a barrier-free path search in a 20x20 grid. The green point is the starting point, the yellow point is the end point, and the blue area is the area searched by the A * algorithm.

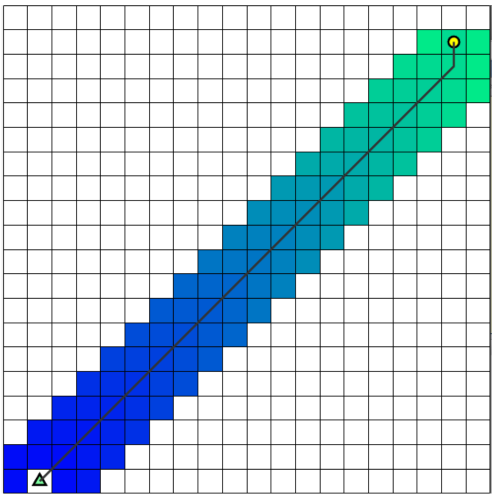

Figure.3 A * algorithm path search in barrier-free area

ISSN: 2414-1895 DOI: 10.6919/ICJE.202102_7(2).0053

4. Simulation and experimental verification

In order to verify the effectiveness of the A* algorithm in the path planning of unmanned vessels, this paper uses electronic charts to select the sea area near Guangzhou for experimental verification.

First, a 60x60 grid map is created based on the electronic chart. A* algorithm and Dijkstra algorithm are used to perform path search in the same chart model, and the search time of the two algorithms is compared. Figure 4 shows Dijkstra's algorithm for path search in environmental seas, and Figure 5 uses A* algorithm for path search. Green is the starting point for unmanned boats, and yellow is the ending point. Through the two pictures, we can intuitively observe that the path distances searched by the two algorithms are the same, and the blue expanded area Dijkstra algorithm is much more than the A* algorithm, that is, the path search time of the A* algorithm is shorter. Through simulation, the path search time of Dijkstra algorithm is 10.7s, and the search time of A* algorithm is 2.3s.

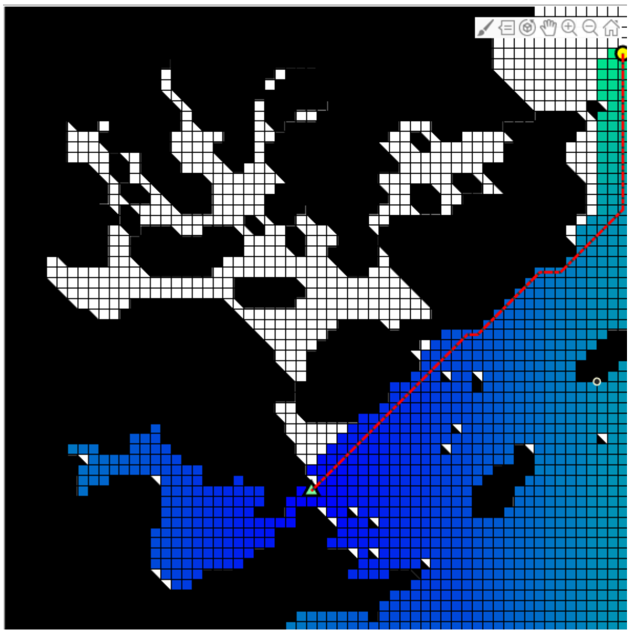
Figure.4 Dijkstra algorithm path search in sea area

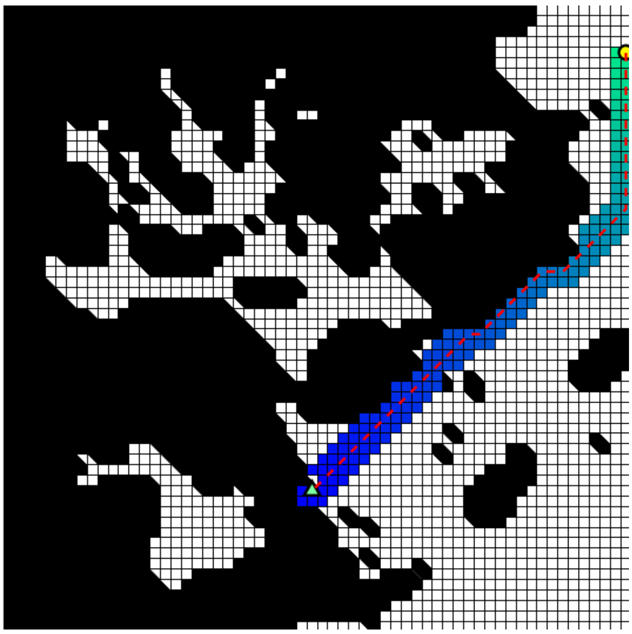
Figure.5 A * algorithm path search in sea area

ISSN: 2414-1895 DOI: 10.6919/ICJE.202102_7(2).0053

5. Conclusions

Aiming at better plan the route of the unmanned vessels, this paper first introduces the Dijkstra algorithm, which is optimized for its algorithm search time and non-intelligence through the A* algorithm, so that the optimal path can be selected faster. Finally, the marine environment model is established and the two algorithms are compared. The simulation results verify the accuracy and speed advantage of the A* algorithm in selecting the path, which provides theoretical support for the development of the unmanned ship path planning algorithm.

Acknowledgments

This work was financially supported by Liaoning Provincial Education Department's 2019 Scientific Research Funding Project (QL201911).

References

[1] Zhixiang Liu, Youmin Zhang, Xiang Yu, et al. Unmanned surface vehicles: An overview of developments and challenges[J]. Annual Reviews in Control, 2016, 41:71-93.

[2] Jia Chen. Research on path planning algorithm of unmanned rescue ship [D]. Wuhan University of technology, 2013.

[3] Kirkpatrick S, Gelatt Jr C D, Vecchi M P. Optimization by simulated annealing[J]. Science, 1983, 220(13): 671~680

[4] SenRao. Research on global path planning technology of surface unmanned vehicle [D]. Harbin Engineering University.

[5] Changlan Xu. Mobile robot design and global path planning [D]. Dalian University of technology, 2006.

[6] Jun Liu, Guangrui Liu. Path planning of mobile robot based on improved ant colony algorithm[J]. Mechanical design and manufacturing, 2010, 000(008):164-165.

[7] Dechter R, Pearl J. Generalized best-first search strategies and the optimality af A*[J]. Journal of the ACM, 1985, 32(3):505-536.

[8] Boschian V, Pruski A. Grid modeling of robot cells: A memory-efficient approach[J]. Journal of Intelligent and Robotic Systems: Theory and Applications, 1993, 8(2):201-223.

[9] Dijkstra, Edsger W. Guarded commands, nondeterminacy and formal derivation of programs[J]. Communications of the Acm, 1975, 18(8):453-457.

[10] Chabini I, Lan S. Adaptations of the A* algorithm for the computation of fastest paths in deterministic discrete-time dynamic networks[J]. IEEE Transactions on Intelligent Transportation Systems, 2002, 3(1): 60-74.