

# Research on the optimum path — Taking the hospital delivery robot as an example

**Guodong Liu**

BSc Computer Science, University College London, Gower Street, London WC1E 6BT, the UK

zcabgli@ucl.ac.uk

**Abstract.** Medical delivery robot refers to the delivery robot used in the medical field. Compared with ordinary delivery robots, the medical delivery robot needs to work in an environment with many people, which means that it needs to deal with many random obstacles at any time. This article will discuss automatically avoiding obstacles, and formulate and compare algorithms to analyze the advantages and disadvantages of automatic road exploration algorithms of different algorithms. In order to accomplish this goal, this paper will use Matlab as the main development tool and use the A-star algorithm and the Euclid algorithm as the main heuristics in the main mathematical model of the program. This program needs to be able to complete the obstacle avoidance task in a map with random size, random position, random shape, and random number of obstacles, and be able to reach the end point from the starting point. In addition, this paper will also discuss the efficiency of the A-star algorithm and the Dijkstra algorithm in obstacle avoidance and route planning, and demonstrate why the A-star algorithm is more efficient by taking time and congestion indicators.

**Keywords:** delivery robot, pathfinding robot, A\* algorithm, Dijkstra algorithm.

## 1. Introduction

With the development of artificial intelligence and intelligent robots, the delivery of robots in various fields, especially in the medical field, becomes more and more common, and more and more robots are used for such work.[1]. In such a work, robots usually only need to send back and forth between two fixed positions, but due to the particularity of their workplace, there may be many random obstacles to prevent the robot from being established. At this time, a qualified obstacle avoidance program is very important [2].

Therefore, in this paper, will aim to show an algorithm that can deal with this situation. In addition, this paper will also discuss the differences in the efficiency of different algorithms in automatic pathfinding and route planning.

This paper mainly uses Matlab as the main development tool. This paper can be used as an algorithm reference and example for automated delivery robots, as well as a reference for the efficiency of the A-star algorithm and the Dijkstra algorithm in different scenarios.

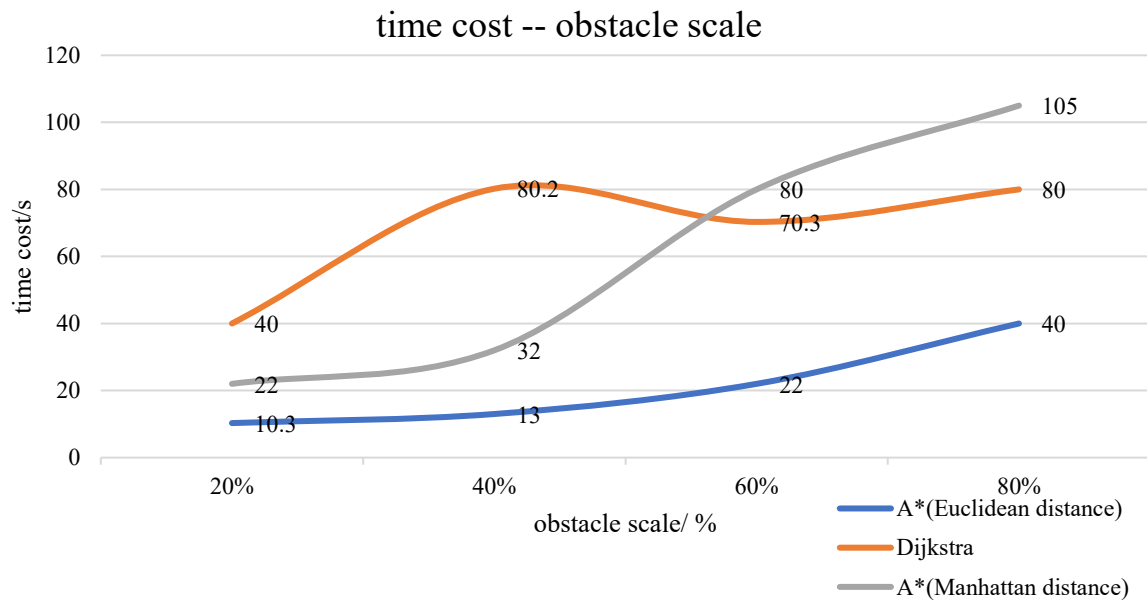


black represents impassable obstacles and white is passable area. In this example picture, the program generates three obstacles, namely two rectangular obstacles and one circular obstacle. Among them, the rectangular obstacle and the circular obstacle block the running path of the delivery robot, so the program avoids them and finds the shortest path from the start point to the end point.

In the main program, it is mainly responsible for the realization and display of the entire program. At the beginning of the program, all existing windows will be cleared first, and then a fixed-size map (400\*400) will be generated and obstacles of random positions and sizes will be generated on the map. Next, use the Euclidean heuristic function to search for the cost of each point, check points around and add the point to the openlist. If a better point is found, add the point of the openlist to the closelist until the end point is added to the open list. A path generated based on the A-star algorithm will be obtained.

After the program finds the shortest path, the program will calculate the time statistics of the shortest path and the proportion of the generated obstacle area in the map as the next research data, and display the calculated path in the form of a drawing.

#### 2.4. Comparison of the results of the two algorithms



**Figure 2.** Experimental data (time cost against obstacle scale).

According to Figure 2, the orange curve represents the performance of Dijkstra's algorithm in the program, and the blue curve represents the performance of the A-star algorithm using the Euclidean heuristic. In general, the A-star algorithm takes less time than the Dijkstra algorithm in terms of time consumption regardless of the size of the obstacle in the map. In more detail, as obstacles occupy a larger area in the map, both algorithms spend more time computing the shortest path. However, the A-star algorithm takes more and more time as the obstacle area increases, while the Dijkstra algorithm takes a stable time between 70 and 80 seconds when the obstacle area becomes larger.

After the program was successfully built and 100 trials were performed, we obtained a set of data describing the efficiency of the A-star algorithm and the Dijkstra algorithm in a fixed-size map to deal with random obstacle scenarios to complete the delivery task. The data contains the best path obtained by the program time, which is used to describe the time consumption of different algorithms to obtain the results. And the proportion of the area of the obstacle randomly generated by the program in the overall map area, which is used to describe the degree of congestion of the map. A larger obstacle means that the obstacle is more likely to block the robot's running route, and the robot has a greater possibility

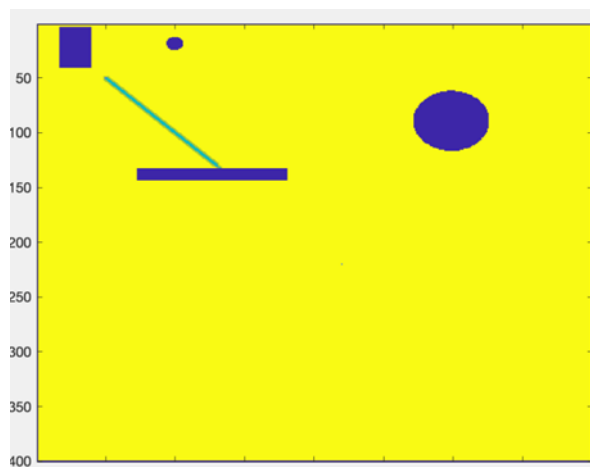
of needing to avoid obstacles. The efficiency ratio of the final two algorithms is obtained through the complexity of time (the larger the efficiency, the more efficient it is to spend less time to complete a more crowded map).

As can be seen from the table 1, as roadblocks gradually occupy more space on the map, the robot has to do more evasion to bypass them, so the robots supported by different algorithms spend more time overall. When the map size and random obstacles are fixed, A-star algorithm is more advantageous than the Dijkstra algorithm in completing point-to-point tasks.

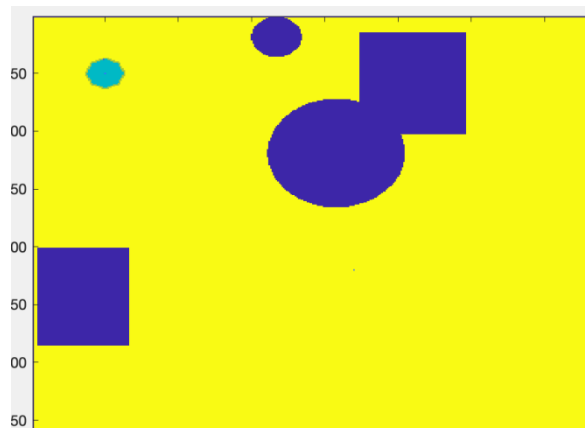
**Table 1.** Points traversed by A-star algorithm and Dijkstra algorithm in finding the path.

	A-star algorithm	Dijkstra algorithm
Open list	752	684
Close list	18794	73703
Total list	271.46	273.22

Continue to analyze the reasons through our previous program, after comparison, it is found that when the two algorithms are looking for the optimal path in the same map, the Dijkstra algorithm often needs to traverse the entire map to find the optimal path. Therefore, compared with the A-star algorithm, the Dijkstra algorithm has data in its close list that is much larger than the A-star algorithm, which makes the Dijkstra algorithm very inefficient and not suitable for use in hospitals [6] [7].



**Figure 3.** A-star algorithm.



**Figure 4.** Dijkstra algorithm.

As can be seen from the figure 3 and figure 4, these two graphs are screenshots from the points traversed in real time while the program is running to find a feasible path. According to figure3, it represents how the A-star algorithm finds the optimal path. When the program is running, it will automatically calculate the lowest cost point around the starting point, until the running path is blocked by obstacles, the path will go back to the starting point and repeat this process [8]. The process until the optimal path is found (reaching the point to bypass the obstacle), so it appears as a thick straight line on the way, which is actually composed of many straight lines composed of single points.

According to Figure 4, this figure shows the logic of the Dijkstra algorithm when traversing the path, traversing all points from the starting point as the origin to the outward circular radiation [9]. So its image is circular.

Further, use a more concrete way to show the operation logic of different algorithms. When the A-star algorithm finds the path, it tends to find the path in a straight line (similar to the best first algorithm), while the Dijkstra algorithm is more accurate when traversing the surrounding points. Like a flood, all the surrounding points will be traversed from the starting point to the starting point. This is mainly because of the heuristic algorithm, which makes the A-star algorithm tend to explore towards the end point when calculating the path, so that the amount of data that the program needs to process is greatly reduced [10].

### 3. Conclusion

The conclusion is that the automatic pathfinding program written in this paper can be well applied in a fixed map with randomly generated and randomly sized obstacles, and the simulated robot can go from the starting point, bypass the obstacles and reach the end point very well. In addition, the A-star algorithm in this scenario is more efficient than the Dijkstra algorithm, because the A-star algorithm does not need to traverse all the points to obtain the optimal path in the process of obtaining the shortest path, while the Dijkstra algorithm will traversing all the points makes Dijkstra's algorithm take more time.

First of all, in terms of programming, the proportion of the size of the obstacles generated by the program in the map may not fully describe the degree of congestion of a map, which may lead to inaccurate final results. In addition, too few statistical data samples may also lead to a large deviation in the final result, because the obstacles generated by the program cannot be guaranteed to be on the running path of the robot, which may lead to the calculation of the shortest path. There is a big difference in time. (Especially the automatic pathfinding program under the A-star algorithm.)

In the future, the map may be set larger and more obstacles may be added, and the space complexity may be used to describe the degree of congestion of the map. And, in the future, this program will continue to work on the development of obstacles that can automatically avoid movable random sizes and generate random positions, so as to be more suitable for the working environment of the robot in the hospital environment.

### References

- [1] www.marketsandmarkets.com. (2023). Delivery Robots Market Size, Growth, Trend and Forecast to 2024. <https://www.marketsandmarkets.com/Market-Reports/delivery-robot-market-263997316.html>.
- [2] Fx361.com. (2022). A Survey of Research on Path Planning Algorithms for Unmanned Express Robots. <https://m.fx361.com/news/2022/0221/16655369.html>
- [3] Emeli, V., Wagner, A.R. and Kemp, C.C. (2012). A Robotic System for Autonomous Medication and Water Delivery. <https://smartech.gatech.edu/bitstream/handle/1853/45009/Medication-Delivery-Tech-Report>.
- [4] Zhihu Column. (2023). Summary of Path Planning Algorithms. <https://zhuanlan.zhihu.com/p/51372134>.
- [5] www.guyuehome.com. (2023). Introduction to Full Coverage Path Planning Algorithms - Gu Yueju. <https://www.guyuehome.com/37569>.

- [6] Kavirayani, S., Uddandapu, D.S., Papasani, A. and Krishna, T.V. (2020). Robot for Delivery of Medicines to Patients Using Artificial Intelligence in Health Care. 2020 IEEE Bangalore Humanitarian Technology Conference (B-HTC). (pp. 1-4). Vijiyapur, India
- [7] Neloy, A.A., Bindu, R.A., Alam, S., Haque, R., Khan, Md.S.A., Mishu, N.M. and Siddique, S. (2020). Alpha-N-V2: Shortest Path Finder Automated Delivery Robot with Obstacle Detection and Avoiding System. Vietnam Journal of Computer Science, 07(04), pp.373–389.
- [8] Zhihu Column. (2023). A\* Algorithm for Path Planning. <https://zhuanlan.zhihu.com/p/54510444>.
- [9] XIAOBAlHAIZAIXIEDAIMA, (2022), Diagram Diagram Diagram Dijkstra Dijkstra Dijkstra - csdn blog. URL: <https://blog.csdn.net/ZER00000001/article/details/125413519>.
- [10] ZHIBIZHANGJIANZUOTIANYA, (2022), Dijkstra Algorithm and A\* Algorithm dijkstra Algorithm and A\* Algorithm\_a\* Algorithm and Dijkstra\_Blog-CSDN Blog. <https://blog.csdn.net/zplai/article/details/105476817>.