Optimization methods for A* and D* algorithms

Leyang Li

Tongji University, 1239 Siping Road, Yangpu District, Shanghai, China

lileyang8322411@gmail.com

Abstract. With the Internet industry's development and the economy's rapid growth, the logistics industry has been overgrowing in recent years. It plays an irreplaceable role in the institutions of economic systems worldwide. However, the A* algorithm and the D* algorithm, which currently dominate the logistics industry's path planning algorithms, still have problems, such as long planning times and long planning paths, and there is much room for optimization. Starting with the conventional A* and D* algorithms, this study improves the planning times of the former by enhancing the heuristic function and the latter by increasing the judgment condition. After verification, the average optimization rate of both improved methods reaches more than 5%, improving the transport efficiency of the logistics industry.

Keywords: A* algorithms, D* algorithms, route planning.

1. Introduction

Online retail sales of tangible things have risen substantially in recent years as a result of the Internet industry's development and the world's rapid economic expansion. At the same time, the sluggish economy (an economy dependent on the fresh produce e-commerce and takeaway industries) is maturing, fuelled by the epidemic over the years [1]. The sluggish economy cannot be separated from the logistics system and is also promoting it. The sluggish economy's demand and its business model's maturity will undoubtedly lead to accelerated economic growth [2].

The development of the express logistics industry, in China, for example, the total amount of social logistics alone grew from 298.0 trillion yuan in 2019 to 335.2 trillion yuan in 2021, an increase of 12.48% year-on-year [3]. However, most of today's logistics algorithms are relatively homogeneous. The main path-planning algorithms that are more commonly used today are the A* and D* algorithms. The shortest path of a network of static roads can be solved using the A* algorithm, which is the most efficient direct search technique [4]. However, the A* algorithm suffers from a slightly longer planning time; For situations when the surrounding environment is unknown or where there are dynamic changes in the surrounding environment, the D* algorithm is a dynamic A* algorithm [5]. However, this algorithm also suffers from the problem of planning long paths in practice, reducing the efficiency of logistics transport to a certain extent [6, 7]. In 2021 alone, logistics companies in China have recorded losses of nearly $30 billion due to logistics delays.

Therefore, in this paper, some improvements are made to the algorithms A* and D* based on path length and planning time, respectively [8]. And this paper plans the route from origin to destination, taking into account the presence of barriers, to improve transport efficiency in the logistics industry. The

A* algorithm's planning time and the D* algorithm's path length are finally optimized by this modification.

2. Literature review

The Stanford Research Institute's Peter Hart, Nils Nilsson, and Bertram Raphael first published the A* search algorithm in 1968 [9]. The A* search algorithm uses a heuristic function to direct the path search process by combining the benefits of the best-first search method with Dijkstra's algorithm:

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

N denotes any vertex, g(n) denotes the actual cost from the starting point to any vertex n, and h(n) stands for a heuristic function that estimates the cost to go from any vertex n to the target location. In each iteration of the algorithm, the node with the smallest value of f(n) (lowest estimation cost) is removed from a priority queue as the node to be traversed next time. This priority queue is often referred to as the OPEN set [10]. Then update their domain node f(n) and g(n) values accordingly and add these domain nodes to the priority queue. When the target node's f(n) value is smaller than any node in the queue, the traversed nodes are placed in a collection known as a CLOSE set. (or until the queue is empty). Since the heuristic function h(n) at the target point is zero, the value of f(n) at the target point is said to be the actual cost of the optimal path. However, the typical A* algorithm's inclusion of the heuristic function h(n) can somewhat increase planning times for big maps, which is precisely the focus of optimization in the algorithm suggested in this study [11].

The D* algorithm is a dynamic A* algorithm, developed from the static A* algorithm and proposed by Stentz in two papers in 1994 and 1995, mainly for robot pathfinding [12]. The D* method is a heuristic path search technique appropriate for situations where the immediate environment is unknown or when the immediate environment is dynamically changing [13]. The D* algorithm searches from the target point towards the starting point, backward propagation, and backward search, in contrast to the A* method, which searches from the starting point to the target point [14]. Similar to A*, the D* algorithm creates two tables, the OPEN and CLOSE tables[15]. The CLOSE table keeps track of nodes that have been visited, whereas the OPEN table stores all nodes that have been generated but not reviewed [16]. First, visit the nearest point in the network that has not been checked and place this point in the OPEN list to be checked. Then find the nearest point to the start point from the OPEN table, find all the children of this point, and put this point into the CLOSE table. Afterward, the children of this point are examined iteratively. Find the distance values of these child nodes from the start point and place the child nodes in the OPEN table. Repeat the above 2 steps until the OPEN table is empty or the target point is found [17].

The traditional D* algorithm also has a bit of a drawback. One of the more obvious issues is that the paths designed by the D* algorithm, which is a point of optimization in the technique suggested in this study, frequently contain a certain amount of path redundancy, i.e., dentures.

3. Research methods

3.1. Introduction to the research methodology

In this paper, the traditional A* algorithm is optimized in terms of planning time, and the D* algorithm is improved in terms of planning path length.

For the improvement of the A* algorithm, as showed in figure 1, the dataset was first constructed, and the A* algorithm before the improvement was on which time detection is performed. Later attempts were made to modify the A* algorithm's heuristic function in order to compare pre- and post-improvement times on various data sets. The build time for each option was recorded, and the option with the best optimization time was chosen as the final solution.

For the improvement of the D* algorithm, as showed in figure 1, the dataset is still constructed first, and the path length detection is performed on the pre-modified D* algorithm. Subsequent attempts were made to modify the source code for the D* algorithm to compare the pre-and post-improvement paths on different data sets（Here, the Manhattan path is used for the calculation, the unit path length is set to 10 for horizontal and vertical, and 14 for diagonal）, The path lengths were then modified and retested again, with the build lengths for each solution recorded, and the final solution with the best-optimized paths was chosen.

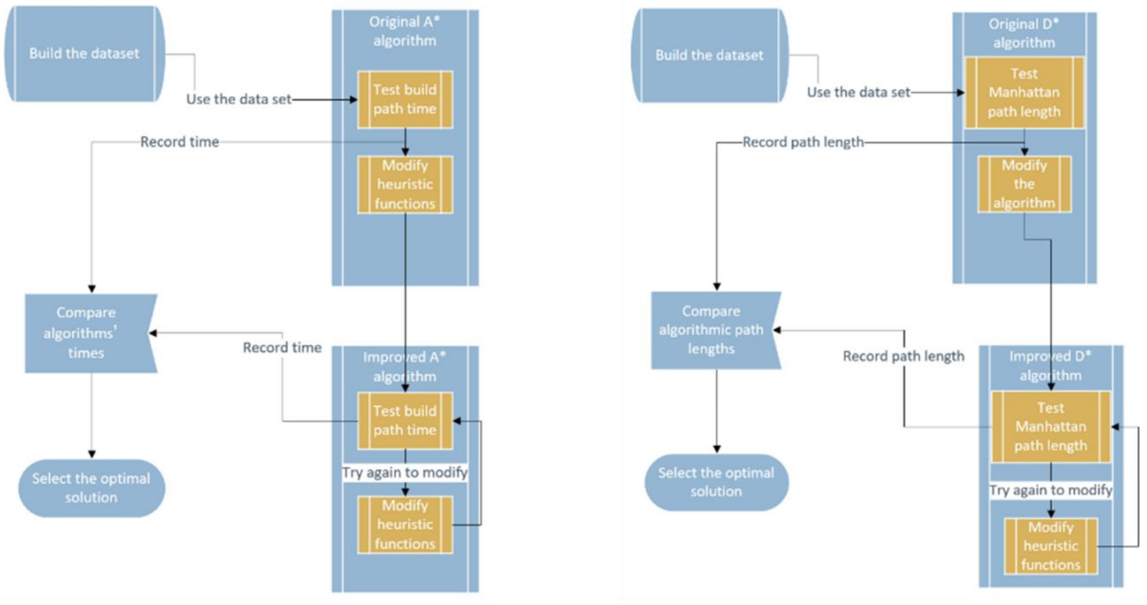


Figure 1. The improvement flowchat of (a) A* algorithm and (b) D* algorithm.

3.2. Introduction to the data set

The datasets for each of the two improved algorithms are described below.

For the improved A* algorithm, the data set contains three sets of self-constructed simulated maps with a two-dimensional list structure, using the number "1" for walkable points and the number "0" for obstacles to simulate a two-dimensional map with obstacles in logistic transport For the improved A* algorithm, the data set contains three sets of self-constructed simulated maps with a two-dimensional list structure, using the number "1" for walkable points and the number "0" for obstacles, to simulate a two-dimensional map with obstacles in logistic transport.

The three data sets are for small, medium, and large maps. As showed in figure 2, the small map being 7*15 in size, starting at (0,0) and ending at (1,14).

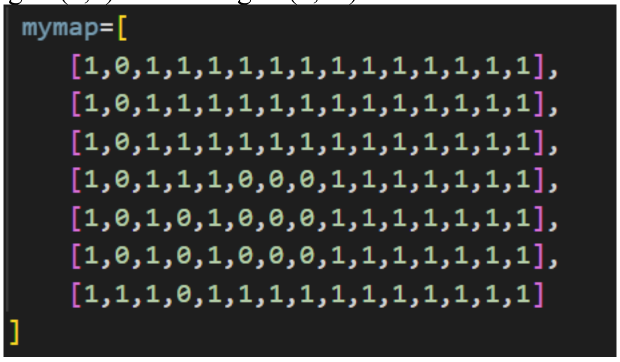
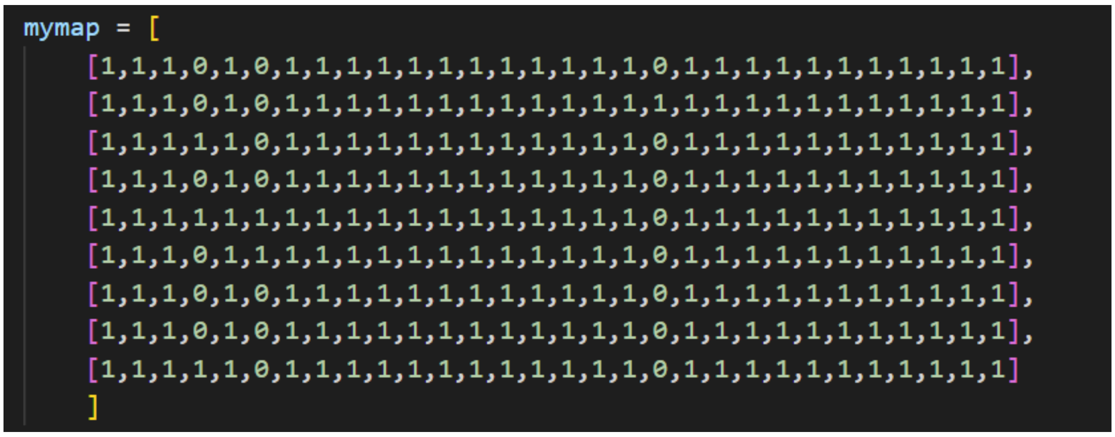

```
mymap=[
    [1,0,1,1,1,1,1,1,1,1,1,1,1,1,1],
    [1,0,1,1,1,1,1,1,1,1,1,1,1,1,1],
    [1,0,1,1,1,1,1,1,1,1,1,1,1,1,1],
    [1,0,1,1,1,0,0,0,1,1,1,1,1,1,1],
    [1,0,1,0,1,0,0,0,1,1,1,1,1,1,1],
    [1,0,1,0,1,0,0,0,1,1,1,1,1,1,1],
    [1,1,1,0,1,1,1,1,1,1,1,1,1,1,1]
]
```

Figure 2. Small map of the improved A* algorithm.

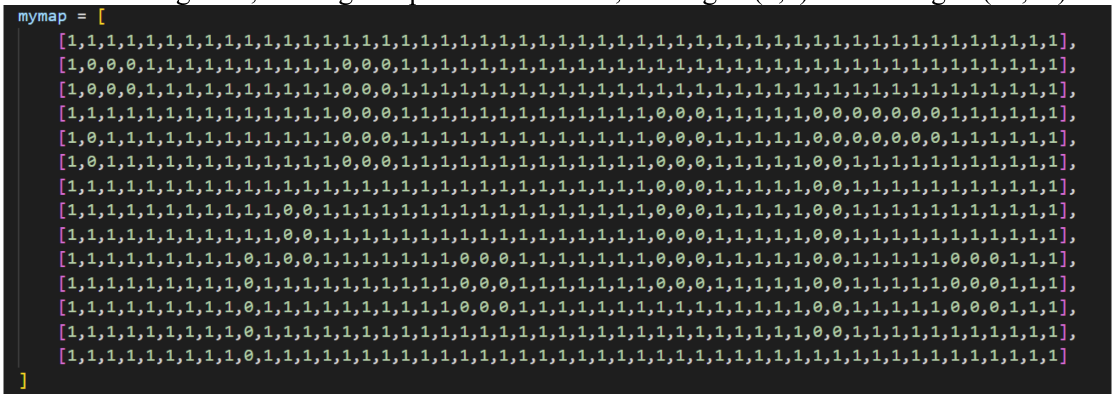
As showed in figure 3, the middle map is 9*30 in size, starting at (0,0) and ending at (8,20).



```
mymap = [
    [1,1,1,0,1,0,1,1,1,1,1,1,1,1,1,1,1,1,0,1,1,1,1,1,1,1,1,1,1,1],
    [1,1,1,0,1,0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1],
    [1,1,1,1,1,0,1,1,1,1,1,1,1,1,1,1,1,1,0,1,1,1,1,1,1,1,1,1,1,1],
    [1,1,1,0,1,0,1,1,1,1,1,1,1,1,1,1,1,1,0,1,1,1,1,1,1,1,1,1,1,1],
    [1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,0,1,1,1,1,1,1,1,1,1,1,1],
    [1,1,1,0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,0,1,1,1,1,1,1,1,1,1,1,1],
    [1,1,1,0,1,0,1,1,1,1,1,1,1,1,1,1,1,1,0,1,1,1,1,1,1,1,1,1,1,1],
    [1,1,1,0,1,0,1,1,1,1,1,1,1,1,1,1,1,1,0,1,1,1,1,1,1,1,1,1,1,1],
    [1,1,1,1,1,0,1,1,1,1,1,1,1,1,1,1,1,1,0,1,1,1,1,1,1,1,1,1,1,1]
    ]
```

Figure 3. Middle map of the improved A* algorithm.

As showed in figure 4, the large map is 14*51 in size, starting at (1,0) and ending at (13,50).

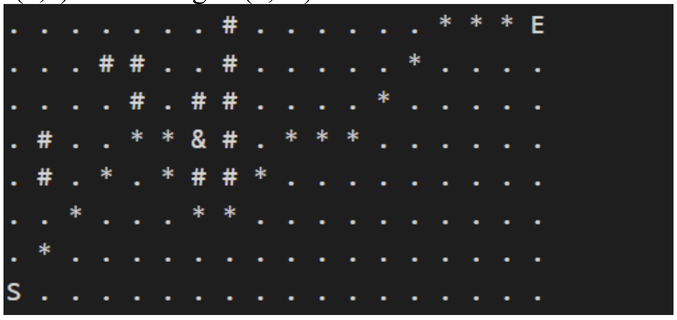


```
mymap = [
    [1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1],
    [1,0,0,0,1,1,1,1,1,1,1,1,1,1,0,0,0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1],
    [1,0,0,0,1,1,1,1,1,1,1,1,1,1,0,0,0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1],
    [1,1,1,1,1,1,1,1,1,1,1,1,1,1,0,0,0,1,1,1,1,1,1,1,1,1,1,1,1,1,0,0,0,1,1,1,1,1,0,0,0,0,0,0,0,1,1,1,1,1,1],
    [1,0,1,1,1,1,1,1,1,1,1,1,1,1,0,0,0,1,1,1,1,1,1,1,1,1,1,1,1,1,0,0,0,1,1,1,1,1,0,0,0,0,0,0,0,1,1,1,1,1,1],
    [1,0,1,1,1,1,1,1,1,1,1,1,1,1,0,0,0,1,1,1,1,1,1,1,1,1,1,1,1,1,0,0,0,1,1,1,1,1,0,0,1,1,1,1,1,1,1,1,1,1,1],
    [1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,0,0,0,1,1,1,1,1,0,0,1,1,1,1,1,1,1,1,1,1,1],
    [1,1,1,1,1,1,1,1,1,1,1,0,0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,0,0,0,1,1,1,1,1,0,0,1,1,1,1,1,1,1,1,1,1,1],
    [1,1,1,1,1,1,1,1,1,1,1,0,0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,0,0,0,1,1,1,1,1,0,0,1,1,1,1,1,1,1,1,1,1,1],
    [1,1,1,1,1,1,1,1,1,0,1,0,0,1,1,1,1,1,1,1,0,0,0,1,1,1,1,1,1,1,0,0,0,1,1,1,1,1,0,0,1,1,1,1,1,0,0,0,1,1,1],
    [1,1,1,1,1,1,1,1,1,0,1,1,1,1,1,1,1,1,1,1,0,0,0,1,1,1,1,1,1,1,0,0,0,1,1,1,1,1,0,0,1,1,1,1,1,0,0,0,1,1,1],
    [1,1,1,1,1,1,1,1,1,0,1,1,1,1,1,1,1,1,1,1,0,0,0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,0,0,1,1,1,1,1,0,0,0,1,1,1],
    [1,1,1,1,1,1,1,1,1,0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,0,0,1,1,1,1,1,1,1,1,1,1,1],
    [1,1,1,1,1,1,1,1,1,0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1]
]
```

Figure 4. Large map of the improved A* algorithm.

For the improved D* algorithm, the data set contains 2 sets of self-constructed simulation maps with a two-dimensional list structure, using the symbol "." for walkable points, the symbol "#" for obstacles, the symbol "+" for the path before the D* algorithm obstacle change, the symbol "*" for the D* algorithm The path after the obstacle change is represented by the symbol "&" for the optimized path after the D* algorithm obstacle change, to simulate a 2D map with obstacles in logistics transport.

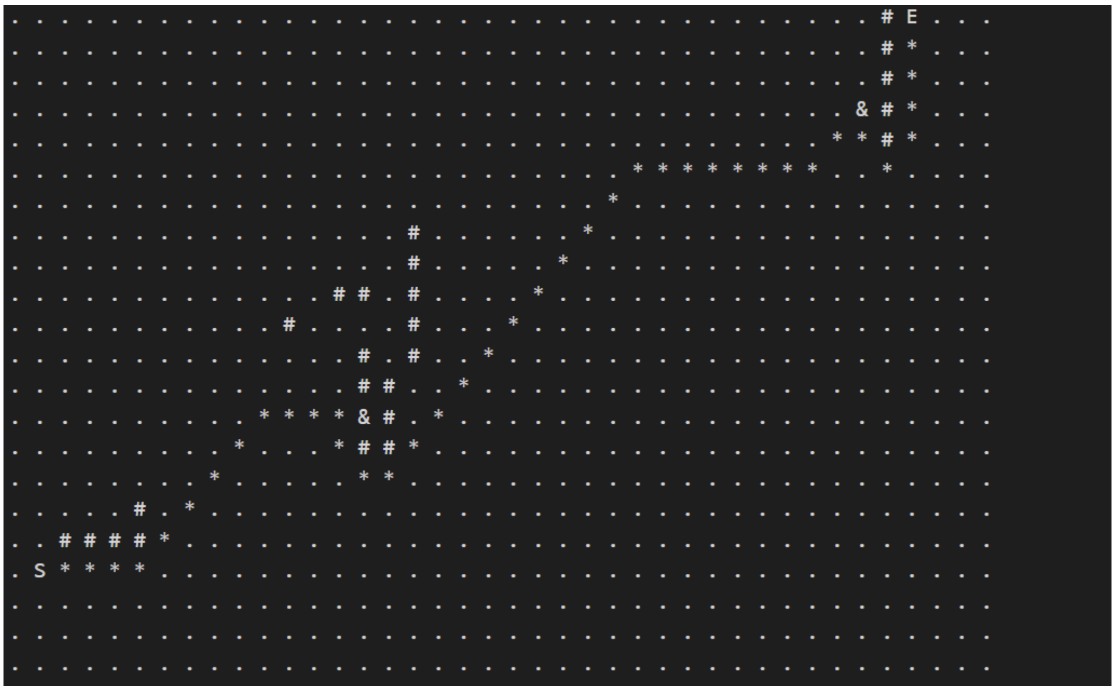
The two sets of data represent the small and large maps. As showed in figure 5, the small map being 8*18 in size, starting at (7,0) and ending at (0,17).



```
. . . . . . . # . . . . . . * * * E
. . . # # . . # . . . . . * . . . .
. . . . # . # # . . . . * . . . . .
. # . . * * & # . * * * . . . . . .
. # . * . * # # * . . . . . . . . .
. . * . . . * * . . . . . . . . . .
. * . . . . . . . . . . . . . . . .
S . . . . . . . . . . . . . . . . .
```

Figure 5. Small map of the improved D* algorithm.

As showed in figure 6, the large map is 22*40 in size and starts at (18,1) and ends at (0,36).



```
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . # E . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . # * . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . # * . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . & # * . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . * * # * . . .
. . . . . . . . . . . . . . . . . . . . . . . . . * * * * * * * * . . * . . . .
. . . . . . . . . . . . . . . . . . . . . . . . * . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . # . . . . . . * . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . # . . . . . * . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . # # . # . . . . * . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . # . . . . # . . . * . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . # . # . . * . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . # # . . * . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . * * * * & # . * . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . * . . . * # # * . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . * . . . . . * * . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . # . * . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . # # # # * . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. S * * * * . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
```

Figure 6. Large map of the improved D* algorithm.

3.3. *Introduction to the algorithm*

The two improved algorithms are described separately below.

The heuristic function of the original A* algorithm has been improved for the upgraded A* method. The formula for the heuristic function in the original A* algorithm is:

$$f(n) = g(n) + h(n) \tag{2}$$

The improved A* algorithm is calculated as follows:

$$f(n) = g(n) + w(n) * h(n) \tag{3}$$

The improvement is multiplied by a w(n) influence factor before h(n), and w(n) is calculated from h(n) by the following formula:

$$w(n) = 1 + \frac{h(n)}{h(n)_i} \tag{4}$$

$h(n)_i$denotes the value of h(n) at the starting point. When h(n) is larger, the farther away from the endpoint, the larger w(n) will be. The faster the algorithm searches the neighborhood, the less accurate the search will be, and the more optimal the path will be. When h(n) is smaller, indicating closer to the endpoint, w(n) is smaller, at which point the algorithm's search speed for the neighborhood decreases and is closer to the original A* algorithm but with a better path.

For the improved D* algorithm, this algorithm adds an extra segment of judgment to the original D* algorithm pathfinding process, i.e., when there are extra steps in the pathfinding process that may occur due to the redundant calculation of the D* algorithm, the algorithm will add an extra step of optimization to avoid such a situation, and its pseudo-code is:

```
Algorithm 1 Improved D* Algorithm
Input: The current direction;
Output: Optimised direction;
1: if there are obstacles above, ahead and below in the direction of travel then
2:     Mark the point as &
3:     Parent of the child node of this point = parent of this point
4:     The point = child node of the point
5:     Output the optimised direction
6: else
7:     Output the original direction
```

3.4. *Introduction to the outcome evaluation methodology*

The methods for evaluating the results of each of the two improved algorithms are described below. The original A* method and the enhanced A* algorithm's path planning times and lengths are measured multiple times on the dataset for the improved A* algorithm in this work. The average is taken for comparison, and a percentage is calculated to show the optimization rate over time.

The datasets for the original D* algorithm and the improved D* algorithm are measured several times for the improved D* algorithm in terms of path planning length and path planning time, respectively. The average is taken for comparison, and the optimisation rate in terms of path length is represented as a percentage.

4. Experiments

4.1. *Experimental setup*

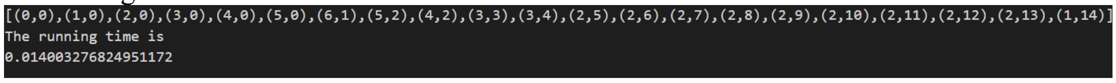
Table 1: The setting of experiment.

Content	Deail
Operating systems	win11
RAM	32G
CPU	AMD Ryzen 7 5800H
Graphics Cards	RTX3070
Coding language	python 3.6

4.2. *Presentation of experimental results*

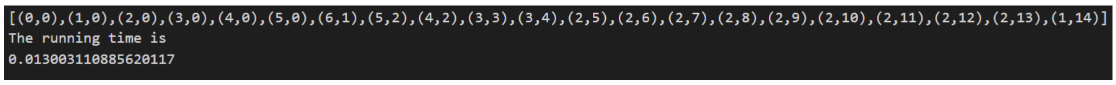
The experimental results of each of the two improved algorithms will be shown below.

The results of the A* algorithm before and after the improvement are shown on the mini-map as figure 7 and figure 8:



```
[(0,0),(1,0),(2,0),(3,0),(4,0),(5,0),(6,1),(5,2),(4,2),(3,3),(3,4),(2,5),(2,6),(2,7),(2,8),(2,9),(2,10),(2,11),(2,12),(2,13),(1,14)]
The running time is
0.014003276824951172
```

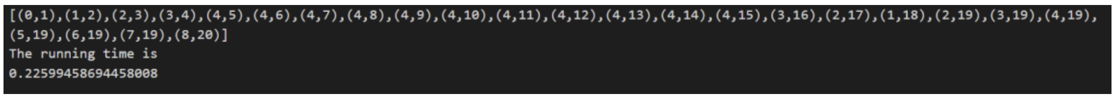
Figure 7. Results of the A* algorithm before improvement on the small map.



```
[(0,0),(1,0),(2,0),(3,0),(4,0),(5,0),(6,1),(5,2),(4,2),(3,3),(3,4),(2,5),(2,6),(2,7),(2,8),(2,9),(2,10),(2,11),(2,12),(2,13),(1,14)]
The running time is
0.013003110885620117
```

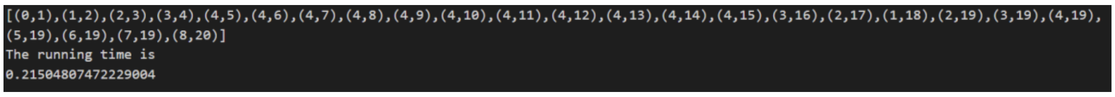
Figure 8. Results of the improved A* algorithm on the small map.

The results of the improved A* algorithm before and after on the medium map are shown as figure 9 and figure 10:



```
[(0,1),(1,2),(2,3),(3,4),(4,5),(4,6),(4,7),(4,8),(4,9),(4,10),(4,11),(4,12),(4,13),(4,14),(4,15),(3,16),(2,17),(1,18),(2,19),(3,19),(4,19),
(5,19),(6,19),(7,19),(8,20)]
The running time is
0.22599458694458008
```

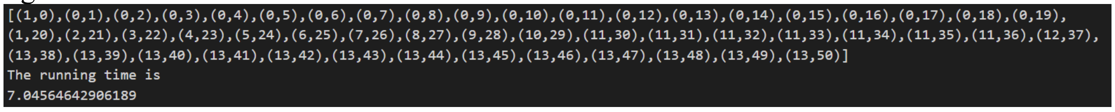
Figure 9. Results of the A* algorithm before improvement on the medium map.



```
[(0,1),(1,2),(2,3),(3,4),(4,5),(4,6),(4,7),(4,8),(4,9),(4,10),(4,11),(4,12),(4,13),(4,14),(4,15),(3,16),(2,17),(1,18),(2,19),(3,19),(4,19),
(5,19),(6,19),(7,19),(8,20)]
The running time is
0.21504807472229004
```

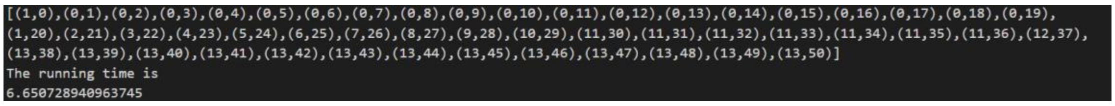
Figure 10. Results of the improved A* algorithm on the medium map.

The results of the improved A* algorithm before and after on the large map are shown as figure 11 and figure 12:



```
[(1,0),(0,1),(0,2),(0,3),(0,4),(0,5),(0,6),(0,7),(0,8),(0,9),(0,10),(0,11),(0,12),(0,13),(0,14),(0,15),(0,16),(0,17),(0,18),(0,19),
(1,20),(2,21),(3,22),(4,23),(5,24),(6,25),(7,26),(8,27),(9,28),(10,29),(11,30),(11,31),(11,32),(11,33),(11,34),(11,35),(11,36),(12,37),
(13,38),(13,39),(13,40),(13,41),(13,42),(13,43),(13,44),(13,45),(13,46),(13,47),(13,48),(13,49),(13,50)]
The running time is
7.04564642906189
```

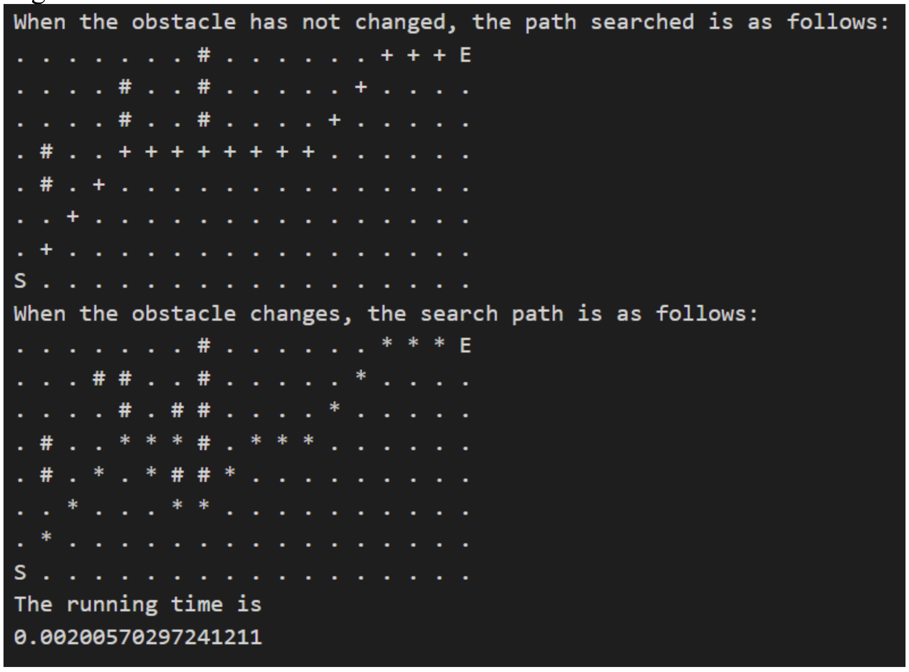
Figure 11. Results of the A* algorithm before improvement on the large map.



```
[(1,0),(0,1),(0,2),(0,3),(0,4),(0,5),(0,6),(0,7),(0,8),(0,9),(0,10),(0,11),(0,12),(0,13),(0,14),(0,15),(0,16),(0,17),(0,18),(0,19),
(1,20),(2,21),(3,22),(4,23),(5,24),(6,25),(7,26),(8,27),(9,28),(10,29),(11,30),(11,31),(11,32),(11,33),(11,34),(11,35),(11,36),(12,37),
(13,38),(13,39),(13,40),(13,41),(13,42),(13,43),(13,44),(13,45),(13,46),(13,47),(13,48),(13,49),(13,50)]
The running time is
6.650728940963745
```

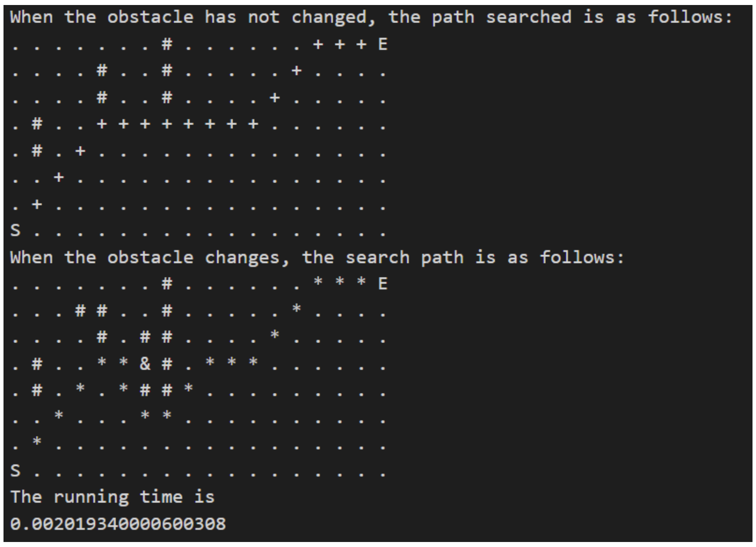
Figure 12. Results of the improved A* algorithm on the large map.

The results of the D* algorithm before and after the improvement are shown on the small map as figure 13 and figure 14:



```
When the obstacle has not changed, the path searched is as follows:
. . . . . . . # . . . . . . + + + E
. . . . # . . # . . . . . + . . . .
. . . . # . . # . . . . + . . . . .
. # . . + + + + + + + + . . . . . .
. # . + . . . . . . . . . . . . . .
. . + . . . . . . . . . . . . . . .
. + . . . . . . . . . . . . . . . .
S . . . . . . . . . . . . . . . . .
When the obstacle changes, the search path is as follows:
. . . . . . . # . . . . . . * * * E
. . . # # . . # . . . . . * . . . .
. . . . # . # # . . . . * . . . . .
. # . . * * * # . * * * . . . . . .
. # . * . * # # * . . . . . . . . .
. . * . . . * * . . . . . . . . . .
. * . . . . . . . . . . . . . . . .
S . . . . . . . . . . . . . . . . .
The running time is
0.00200570297241211
```

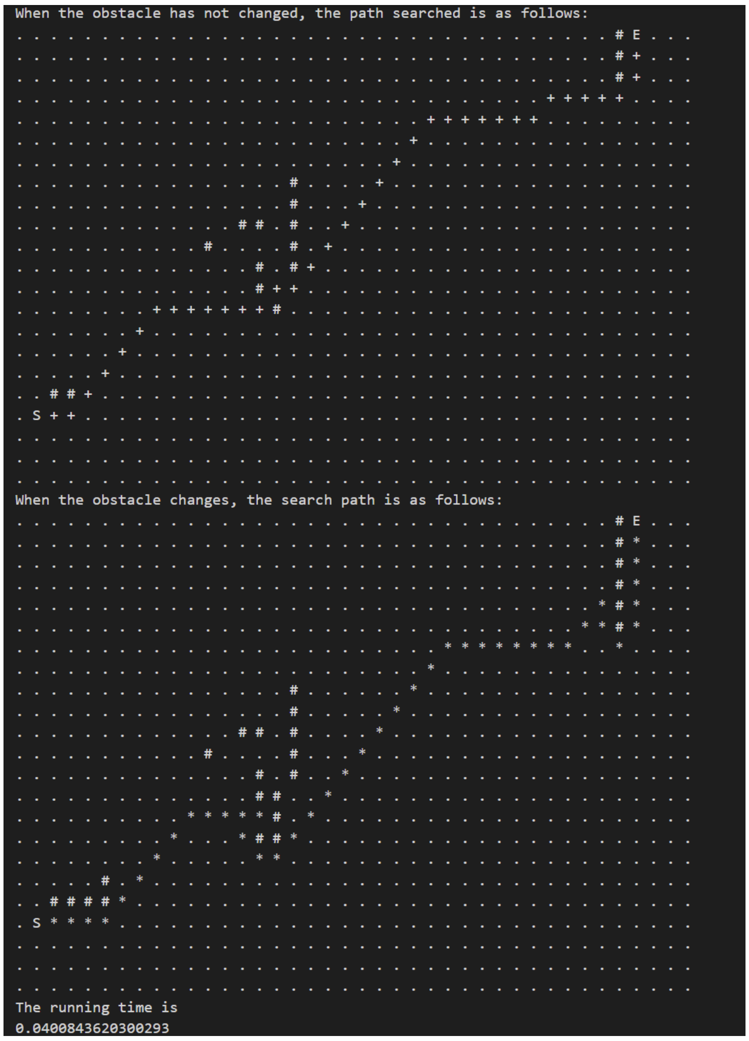
Figure 13. Results of the D* algorithm before improvement on the small map.



```
When the obstacle has not changed, the path searched is as follows:
. . . . . . . # . . . . . . + + + E
. . . . # . . # . . . . . + . . . .
. . . . # . . # . . . . + . . . . .
. # . . + + + + + + + + . . . . . .
. # . + . . . . . . . . . . . . . .
. . + . . . . . . . . . . . . . . .
. + . . . . . . . . . . . . . . . .
S . . . . . . . . . . . . . . . . .
When the obstacle changes, the search path is as follows:
. . . . . . . # . . . . . . * * * E
. . . # # . . # . . . . . * . . . .
. . . . # . # # . . . . * . . . . .
. # . . * * & # . * * * . . . . . .
. # . * . * # # * . . . . . . . . .
. . * . . . * * . . . . . . . . . .
. * . . . . . . . . . . . . . . . .
S . . . . . . . . . . . . . . . . .
The running time is
0.002019340000600308
```

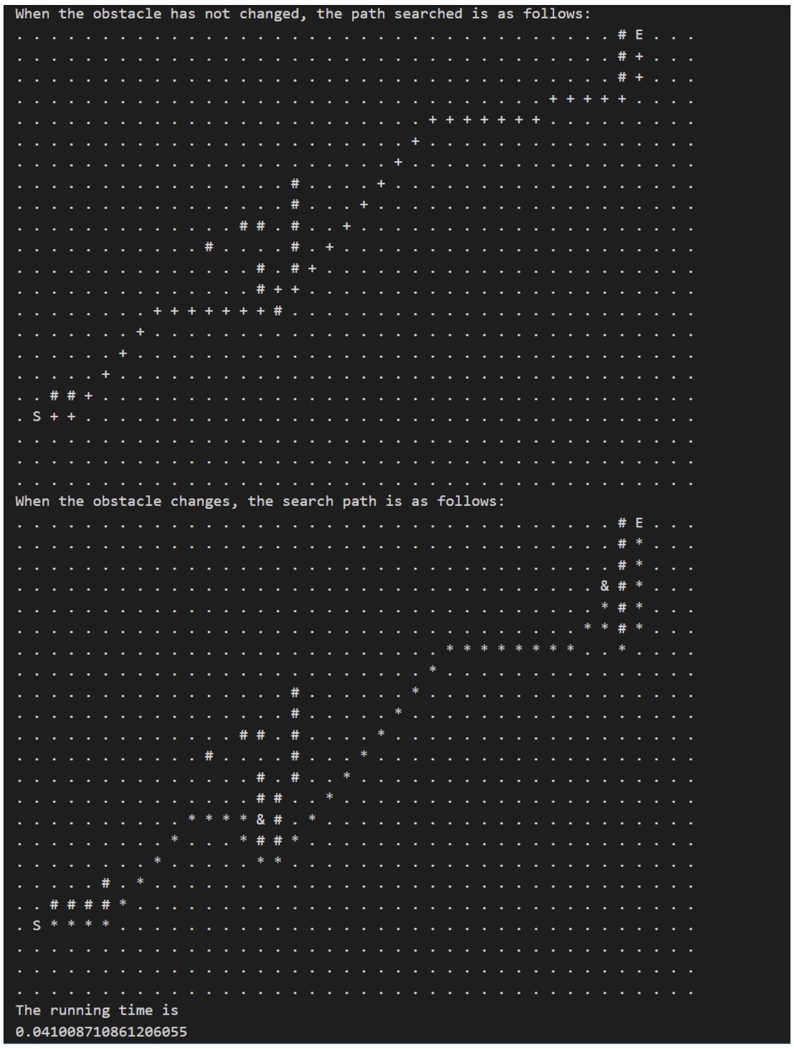
Figure 14. Results of the improved D* algorithm on the small map.

The results of the improved D* algorithm before and after on the large map are shown as figure 15 and figure 16:



```
When the obstacle has not changed, the path searched is as follows:
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . # E . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . # + . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . # + . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . + + + + + . . . .
. . . . . . . . . . . . . . . . . . . . . . . . + + + + + + + . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . + . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . + . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . # . . . . + . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . # . . . + . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . # # . # . . + . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . # . . . . # . + . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . # . # + . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . # + + . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . + + + + + + + # . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . + . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . + . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . + . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . # # + . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. S + + . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
When the obstacle changes, the search path is as follows:
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . # E . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . # * . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . # * . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . # * . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . * # * . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . * * # * . . .
. . . . . . . . . . . . . . . . . . . . . . . . . * * * * * * * * . . * . . . .
. . . . . . . . . . . . . . . . . . . . . . . . * . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . # . . . . . . * . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . # . . . . . * . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . # # . # . . . . * . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . # . . . . # . . . * . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . # . # . . * . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . # # . . * . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . * * * * * # . * . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . * . . . * # # * . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . * . . . . . * * . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . # . * . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . # # # # * . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. S * * * * . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
The running time is
0.0400843620300293
```

Figure 15. Results of the D* algorithm before improvement on the large map.



```
When the obstacle has not changed, the path searched is as follows:
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . # E . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . # + . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . # + . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . + + + + + . . . .
. . . . . . . . . . . . . . . . . . . . . . . . + + + + + + + . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . + . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . + . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . # . . . . + . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . # . . . + . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . # # . # . . + . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . # . . . . # . + . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . # . # + . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . # + + . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . + + + + + + + # . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . + . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . + . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . + . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . # # + . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. S + + . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
When the obstacle changes, the search path is as follows:
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . # E . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . # * . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . # * . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . & # * . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . * # * . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . * * # * . . .
. . . . . . . . . . . . . . . . . . . . . . . . . * * * * * * * * . . * . . . .
. . . . . . . . . . . . . . . . . . . . . . . . * . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . # . . . . . . * . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . # . . . . . * . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . # # . # . . . . * . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . # . . . . # . . . * . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . # . # . . * . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . # # . . * . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . * * * * & # . * . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . * . . . * # # * . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . * . . . . . * * . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . # . * . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . # # # # * . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. S * * * * . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
The running time is
0.041008710861206055
```

Figure 16. Results of the improved D* algorithm on the large map.

4.3. *Comparison of experimental results analysis*

The experimental outcomes of the two enhanced algorithms will each be studied and contrasted in the table that follows.

Table 2: Comparison and analysis of time result before and after A* algorithm improvement.

	Small map（s）	Medium map（s）	Large map（s）
Before improvements	0.01400	0.22599	7.04565
After improvements	**0.01300**	**0.21504**	**6.65073**
Optimisation rate	7.143%	4.845%	5.605%

Table 3: Comparison and analysis of path length before and after D* algorithm improvement.

	Small map	Large map
Before improvements	234	504
After improvements	220	476
Optimisation rate	5.983%	5.556%

It can be seen that the optimisation of the A* algorithm for time planning and the D* algorithm for path planning both achieved optimisation rates of roughly 5%.

5. Conclusion

This paper begins with a systematic introduction to the current development of the express logistics industry and mentions the more popular A* algorithm and D* algorithm for path planning in the logistics industry. The proposed and content of the A* and D* algorithms are then summarised in general terms, and some shortcomings of the two algorithms are presented, i.e. the A* algorithm suffers from the problem of long planning times, and the D* algorithm from the problem of redundant planning paths.

This paper then makes some improvements to the A* algorithm and the D* algorithm at the algorithm level, tests the improved algorithms on a two-dimensional map dataset constructed by the authors themselves, and then compares and analyses the results, concluding that the optimisation of the A* algorithm for time planning and the D* algorithm for path planning both achieve an optimisation rate of approximately 5%.

However there are still some shortcomings in this paper. Firstly, there is some optimisation compared to the original algorithm, but the optimisation rate still needs to be improved. Secondly, the improved A* algorithm is likely to have the side effect of sacrificing some path planning accuracy on very large maps, and the improved D* algorithm will sacrifice some path planning time, both of which shortcomings can be considered for optimisation in subsequent work.

References

[1] L. Wang, S.-j. Lee, X.-f. Wu, B.-l. Liu, and J.-h. Xiao, "Contemporary Logistics in China," Springer Science and Business Media LLC, 2020.
[2] A. Amling and P. J. Daugherty, "Logistics and distribution innovation in China," International Journal of Physical Distribution & Logistics Management, vol. 50, no. 3, pp. 323-332, 2020.
[3] M. Giuffrida, R. Mangiaracina, A. Perego, and A. Tumino, "Cross-border B2C e-commerce to Greater China and the role of logistics: a literature review," International Journal of Physical Distribution & Logistics Management, 2017.
[4] D. O. Pugas, M. Somantri, and K. I. Satoto, "Pencarian Rute Terpendek Menggunakan Algoritma Dijkstra dan Astar (A*) pada SIG Berbasis Web untuk Pemetaan Pariwisata Kota Sawahlunto," Transmisi, vol. 13, no. 1, pp. 27-32, 2011.
[5] W. E. Wong, V. Debroy, R. Gao, and Y. Li, "The DStar method for effective software fault localization," IEEE Transactions on Reliability, vol. 63, no. 1, pp. 290-308, 2013.
[6] L. Qin, J. X. Yu, and L. Chang, "Diversifying top-k results," arXiv preprint arXiv:1208.0076, 2012.
[7] C. Hocaoglu and A. C. Sanderson, "Planning multiple paths with evolutionary speciation," IEEE transactions on evolutionary computation, vol. 5, no. 3, pp. 169-191, 2001.
[8] Z. Wang, X. Xiang, J. Yang, and S. Yang, "Composite Astar and B-spline algorithm for path planning of autonomous underwater vehicle," in 2017 IEEE 7th International Conference on Underwater System Technology: Theory and Applications (USYS), 2017: IEEE, pp. 1-6.
[9] C. Thorpe and L. Matthies, "Path relaxation: Path planning for a mobile robot," in OCEANS 1984, 1984: IEEE, pp. 576-581.
[10] Y. Fan, F. Deng, and X. Shi, "Multi-robot task allocation and path planning system design," in 2020 39th Chinese Control Conference (CCC), 2020: IEEE, pp. 4759-4764.
[11] L. Chang, X. Feng, X. Lin, L. Qin, and W. Zhang, "Efficient graph edit distance computation and verification via anchor-aware lower bound estimation," arXiv preprint arXiv:1709.06810, 2017.
[12] H. Li, N. Xu, G. Liu, and J. Zhang, "An optimized 3D Astar algorithm for multi-layer PCB automatic routing," in 2021 IEEE International Conference on Consumer Electronics-Taiwan (ICCE-TW), 2021: IEEE, pp. 1-2.
[13] S. Kambhampati and L. Davis, "Multiresolution path planning for mobile robots," IEEE Journal on Robotics and Automation, vol. 2, no. 3, pp. 135-145, 1986.
[14] K. Sugihara and J. Smith, "Genetic algorithms for adaptive motion planning of an autonomous mobile robot," in Proceedings 1997 IEEE International Symposium on Computational

Intelligence in Robotics and Automation CIRA'97.'Towards New Computational Principles for Robotics and Automation', 1997: IEEE, pp. 138-143.
[15] Z. Xu, X. Liu, and Q. Chen, "Application of improved Astar algorithm in global path planning of unmanned vehicles," in 2019 Chinese Automation Congress (CAC), 2019: IEEE, pp. 2075-2080.
[16] S. Shekhar, A. Kohli, and M. Coyle, "Path computation algorithms for advanced traveller information system (ATIS)," in Proceedings of IEEE 9th International Conference on Data Engineering, 1993: IEEE, pp. 31-39.
[17] Z. Wang and X. Xiang, "Improved astar algorithm for path planning of marine robot," in 2018 37th Chinese Control Conference (CCC), 2018: IEEE, pp. 5410-5414.