ISSN 2616-5775 Vol. 5, Issue 5: 97-102, DOI: 10.25236/AJCIS.2022.050513

Improvement and Application of Dijkstra Algorithms

Heming Bing[1], Lu Lai[2]

[1]*Department of Science, Shenyang Aerospace University, Liaoning, Shenyang, 110136, China*
[2]*Department of Mathematics, Xinyu University, Xinyu, Jiangxi, 338004, China*

Abstract: *Taking urban express traffic as an example, this paper lists two optimization improvements of Dijkstra algorithm. One is to improve the calculation method of the algorithm to make the calculation more directional, so as to reduce the total amount of calculation. One is in the form of a graphical model, reducing the number of nodes to optimize the calculation process. By comparing the calculation results with the complexity of the traditional algorithm, the effectiveness of the improved method is proved, the complexity of the algorithm is reduced, and the actual work efficiency is improved. It improves the application ability and realizability of Dijkstra algorithm in practical problems.*

Keywords: *Dijkstra algorithm; path programming; Optimization and improvement; Application advantages*

1. Introduction

The famous Dutch computer scientist E.W.Dijkstra proposed the far-reaching Dijkstra shortest path algorithm in 1958[1]. It is widely used in traffic route planning [2], logistics transportation and distribution[3], network optimization[4], robot navigation[5] and other fields. Nowadays, with the popularization of online shopping, the logistics industry continues to develop, and it has quickly become a key industry that promotes Chinese economic development and employment growth. In this case, how to plan logistics transportation routes and make the delivery time as short as possible becomes more and more important. Therefore, the Dijkstra method, which solves the shortest path finding problem, has also received more and more attention. With the development of Dijkstra algorithm, many improved calculation methods have appeared, such as table calculation method, storage structure optimization, ellipse search model and rectangular search model. Shortest path algorithms such as SPFA algorithm, Floyd algorithm, Johnson algorithm and A* algorithm are also constantly developing.

The Dijkstra algorithm is an exhaustive traversal algorithm. The way to find the shortest path in the graph is to record all routes in the form of layered diffusion under the premise of ensuring that the routes are not repeated[6], calculate the length of all routes, and compare the shortest Path[7]. Most of the calculated routes are not included in the final path planning scheme[8], which generates a lot of useless calculations and reduces the efficiency of the algorithm. Consequently, the magnitude of the complexity of Dijkstra's algorithm is the square of the number of nodes[9], so it is very inefficient in solving practical problems[10].

Through the analysis of actual problems, this paper proposes two methods to reduce the amount of calculation. One is that when calculating the minimum distance between a certain node and the starting point, the distance from the end point is also considered, so that the exploration is carried out in a general direction toward the end point. Another method is to use an ellipse model to limit the search range, thereby reducing useless calculations.

2. Dijkstra algorithm

2.1. Case Analysis

Urban public transportation arteries and branch roads[11] are complicated. This article takes the YTO Express around Jinjiang District as an example to prove the role of the improved Dijkstra algorithm in practical applications. By investigating some express transportation in Jinjiang District of Chengdu, we have obtained the transportation routes of YTO in this area. Ignore the influence of some factors to study the length of the path. The logistics distribution center can treat the entire urban transportation network as a plan to calculate the shortest path, thereby saving transportation costs and time[12].

ISSN 2616-5775 Vol. 5, Issue 5: 97-102, DOI: 10.25236/AJCIS.2022.050513

Description of the model symbols in the text: 1: YTO Express in the south section of Qingheli 2: Xinzhuang North East 5, Zhulin Village, Guojiaqiao North Street 3: Gongnongyuan Street Store 4: Sichuan University Business Office 6: Jiuyanqiao Business Office 7: YTO Express Wu Hou Yi 8: YTO Express in Jinjiang District 9: Haijiao City Business Office 10: Shuinianhe Road Store 11: Dianjiang Taidong No. 2 Alley and Dianjiang Taidong Street intersection 50 meters east of 12: Yuantong Express Shangdu Branch 13: Hengjiulong Lane 14: Jinjiang District Second Division 15: Chenghua District Xiangheli 131 16: Jindong Road No. 314 17: Shuinian Hebei Third Street 18: Chenghua Second Division. Due to the complexity of traffic, the routes extend in all directions, and only part of the routes are selected as references in this article. As shown in Figure 1.

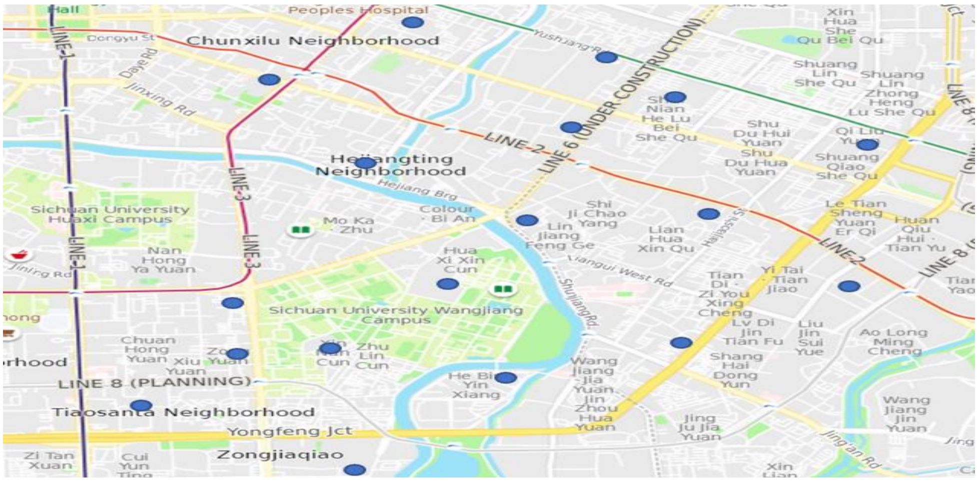

Figure 1: Actual map

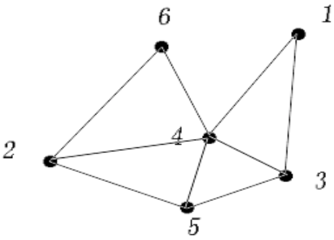

Figure 2: Roadmap 1 to 2

1-3:2.2 km; 1-4:2.6 km; 3-4:3.3 km; 3-5:3.5 km; 4-5:1.5 km; 4-2:4.4 km; 4-6:2.1 km; 6-2:5.1 km; 5-2:2.6 km

Since the actual road sections are all curved roads, straight lines are used instead. v represents the point n.v_n represents the point n.

Calculate the shortest path from 1 to 2:

(1) Mark v_1 point with p, $p(v_1) = 0$, Other T labels, $T(v_i) = +\infty(i = 2, \cdots, 6)$

(2) Observe v_1:$T(v_3) = min\{T(v_3), P(v_1) + l_{13}\} = min\{+\infty, 0 + 2.2\} = 2.2$

$$T(v_4) = min\{T(v_4), P(v_1) + l_{14}\} = min\{+\infty, 0 + 2.6\} = 2.6$$

(3) $T(v_3)$ is the smallest, let $P(v_3) = 2.2$, path(v_1,v_3).

(4) Observe v_3:$T'(v_4) = min\{T(v_4), P(v_3) + l_{34}\} = min\{2.6, 2.2 + 3.3\} = 2.6$

$$T(v_5) = min\{T(v_5), P(v_3) + l_{35}\} = min\{+\infty, 2.2 + 3.5\} = 5.7$$

(5) $T(v_4)$ is the smallest, let$P(v_4) = 2.6$, path(v_1, v_4).

(6) Observe v_4:$T(v_2) = min\{T(v_2), P(v_4) + l_{24}\} = min\{+\infty, 2.6 + 4.4\} = 7$

$$T'(v_5) = min\{T(v_5), P(v_4) + l_{45}\} = min\{5.7, 2.6 + 1.5\} = 4.1$$

$$T(v_6) = min\{T(v_6), P(v_4) + l_{46}\} = min\{+\infty, 2.6 + 2.1\} = 4.7$$

(7) $T'(v_5)$ is the smallest, let$P(v_5) = 4.1$,path(v_4, v_5).

ISSN 2616-5775 Vol. 5, Issue 5: 97-102, DOI: 10.25236/AJCIS.2022.050513

(8) Observe v_5:$T'(v_2) = min\{T(v_2), P(v_5) + l_{25}\} = min\{7,4.1 + 2.6\} = 6.7$

(9) $T(v_6)$ is the smallest, let$P(v_6) = 4.7$, path(v_4, v_6).

(10) Observev_6:$T''(v_2) = min\{T'(v_2), P(v_6) + l_{26}\} = min\{6.7,4.7 + 5.1\} = 6.7$

The shortest path that leads to 1 to 2 is $1 \to 4 \to 5 \to 2$, The shortest distance is 6.7 km

3. Algorithm improvement

3.1. Node optimization direction

When there are too many temporary nodes, it is obvious that there is no need to compare all nodes in each expansion process, that is, some nodes can be searched less, and the calculation steps can be reduced. Here we focus on optimizing how to correctly reduce the number of unrelated nodes. As shown in Figure 3.

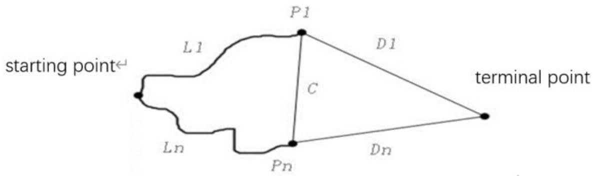

Figure 3: Proof example diagram

3.1.1. Proof of method

$L1$ is the shortest path length between $P1$ and the starting point, Ln is the shortest path length between Pn and the starting point, $D1$ is the straight-line distance between point $P1$ and the end point, Dn is the straight-line distance between node Pn and the end point, and the line between $P1$ and Pn is C.

The path selection principle of this algorithm is: When$L1 + D1 > Ln + Dn$, take Pn as a permanent marker node, when $L1 + D1 < Ln + Dn$, take Pn as a permanent marker node. Thus, the destination is also considered while determining the path. In the above improvement, a prerequisite for the principle of permanent label node selection is that $L1$ is the shortest distance from $P1$ to the starting point, that is, the shortest path value of $P1$ will not be modified. Proof:

Prove $Ln + C > L1$ By $L1 + D1 < Ln + Dn$

Due to $Ln + Dn$>L1+D1

And $Dn \leq C + D1$ so $Ln + C$>L1 Q.E.D

3.1.2. Example calculation is shown in Figure 4

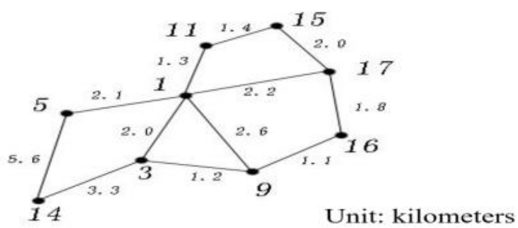

Figure 4: 14 to 15 route map

In practice, most road sections are curved roads, and straight lines are used instead.

Dn is the linear distance from point n to the end point 15. $D5$:4.3km, $D3$:4.0km $D1$:2.1km, $D9$:3.0km, $D11$:1.2km, $D16$:3.2km, $D17$:1.6km

$L(n - p)$ $is\ the\ length\ of\ the\ path\ from\ point\ n\ to\ point\ p.$ When calculating the shortest path between 14 and 15 with the above method, we can get:

(1) Obtain point 3 as a permanent marking point.

(2) Obtain point 1 as a permanent marking point.

(3) Obtain point 11 as a permanent marking point.

(4) Finally, it is concluded that the shortest path from 14 to 15 is:14→3→1→11→15

3.1.3. Conclusion and limitations

The advantage of this method is that when the calculation reaches point 3, it is the next step of permanent labeling. When labeling according to the traditional Dijkstra method, because 2.0 is greater than 1.2, point 9 is used as a temporary label, and the paths from 9 to 1 and 9 to 16 are calculated. The straight-line distance added to the end point after optimization is more directional, and 1 is a permanent labeling point, and temporary nodes will not be calculated too much. At present, most degree path planning is to take a shortest path between two points, often using Dijkstra algorithm and its improved algorithm. However, in actual urban road conditions, the road conditions are changeable, and there will be more factors to be considered in path planning, which is also the limitation of the improvement.

3.2. Graphical model optimization

3.2.1. Round model

Graphical mathematical model optimization refers to optimizing Dijkstra's algorithm according to graphic properties by establishing a graphic mathematical model. The traditional algorithm uses a circular search range to find the shortest path, and takes the distance from the starting node to the target node as the radius of the concentric circle.

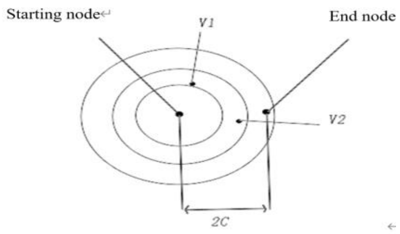

Figure 5: Circular model diagram

We assume that the starting point is the center of the circle. When executing the algorithm, we first search from the starting point and diverge outward. V1 is the shortest for labeling, and then V1 is the origin, and then a circle is drawn to find the next closest point for labeling. Until the end node, find the shortest path.

3.2.2. Ellipse model

The circular model search is accurate, but it takes a long time to determine the search range by drawing a circle when calculating any node. It lacks real-time performance in actual application. In order to solve the problems of the circular model, this paper proposes an ellipse optimization method. The ellipse model takes the start point and the end point as the two focal points of the ellipse, then calculates the size of the ellipse, and discards the nodes outside the search ellipse. This search method only needs to determine the search range once, and can reduce more irrelevant nodes, and the efficiency is also improved.

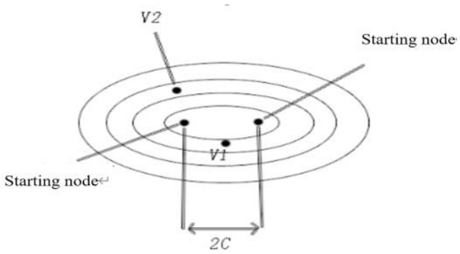

Figure 6: Ellipse model diagram

(1) Determine the starting point O and ending point D, and calculate the distance between the two points;

(2) Taking the starting point and ending point as the focus, and taking the length of $r|OD|$ as the long axis of the ellipse, the search area of the ellipse model is calculated. In this paper, let r be equal to

$\sqrt{2}$.

(3) Determine whether the node is within the search ellipse. Let Vi be the node to be judged, judge whether the distance from the start point to Vi and the distance from the end point to Vi satisfy $|OVi| + |ViD| < 2a$, if so, keep the node, and then search If it is not satisfied, exclude the node. (In order to avoid the situation that the search range is too small and the shortest path cannot be found or the wrong way is found, the value of r should be as large as possible to ensure more accurate results)

(4) Use the traditional algorithm to search the nodes reserved in the previous step, find the shortest path OD, and the algorithm ends.

As shown in Figure 7.

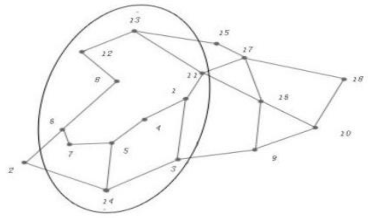

Figure 7: The actual road map of the ellipse

When calculating the shortest path from point 14 to point 13, we take point 14 and point 13 as the focus and the long axis $2a = 2\sqrt{2c}$ to establish an elliptical trajectory. According to the above optimization method, the points outside the ellipse can be directly excluded, and the traditional algorithm is used to calculate the shortest path for the path remaining in the ellipse.

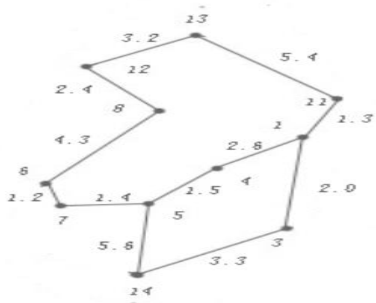

Figure 8: Ellipse optimization diagram

The advantage of its improvement is that it can directly reduce most of the temporary punctuation. Changed from the original multiple paths to 3 paths. As shown in figure 8. Calculated according to the traditional Dijkstra algorithm, the shortest path is $14 \rightarrow 3 \rightarrow 1 \rightarrow 11 \rightarrow 13$.

3.2.3. Model optimization conclusion

The ellipse model reduces part of the nodes, and directly makes the calculated nodes determine the inside of the ellipse. The optimization of Dijkstra's algorithm, in the optimization of the graph model, is mainly to directly reduce some unnecessary computing nodes by optimizing the search range, so that the algorithm has fewer steps and higher efficiency.

4. Conclusion

Compared with the traditional method, the route planning scheme calculated in this paper can save the time of transportation and make the transportation work more efficient. In 2.1, using the traditional algorithm to calculate the shortest path problem of an undirected graph with 6 nodes requires 9 calculations, while using two improved algorithms to calculate the shortest path problem of an undirected graph with 9 and 18 nodes respectively, only Need to calculate 7 times, 9 times. To a certain extent, the total amount of calculation is reduced. Compared with the entire urban transportation network and network routes, the number of computing nodes in this paper is relatively small. When the computing scale is larger, the efficiency improvement effect after optimization will be more obvious. And through the real-time management and maintenance of the transportation network, the network operation efficiency and stability have been improved to a certain extent.

ISSN 2616-5775 Vol. 5, Issue 5: 97-102, DOI: 10.25236/AJCIS.2022.050513

References

[1] Dijkstra E W. A Note on two problems in connection with graphs [J]. Nume Mathematatik, 1959,1:269-271.
[2] Hongbo Wu, Yingjie Wang, Xiaoxiao Yang. Analysis of Urban Traffic Route Based on Dijkstra Algorithm Optimization [J]. Journal of Beijing Jiaotong University, 2019, 43(04): 116-121+130
[3] Boli Huang. Research on Passenger's Optimal Travel Plan Planning Based on Highway-Rail Multimodal Transport Mode [C]. South China University of Technology, 2020.
[4] Jianqi Lu, Jianguo Nan, Xue Li. Research and Application of Improved Greedy Algorithm in UAV Networking [J]. Journal of Air Force Engineering University (Natural Science Edition), 2020, 21(02): 41-46.
[5] Desheng Feng. Research on Mobile Robot Navigation System Based on ROS [C]. Qilu University of Technology, 2021.
[6] Peng Xu, Gang Cheng, Minyi Li. Dijkstra Optimization and Application of Algorithm in GIS [J]. Beijing Surveying and Mapping, 2021, (06): 737-740.
[7] Wenqing Zheng, Danyang Lu. Dijkstra Research on Algorithm Optimization of Logistics System [J]. Shandong University of Science and Technology, 2018, (05),158-159.
[8] Dujuan Huang, Tianyuan Zhang. Research on Township Location Selection of Smart Express Cabinet Based on Dijkstra Algorithm [J]. Journal of Liaoning University of Technology (Natural Science Edition), 2020, 40(04): 259-263.
[9] Bandy JA, Marty USR. Graph Theory with Applications. New York: The Macmillan Press ltd.1976. 1-22.
[10] Wenxiong Kang, Yaozhao Xu. Hierarchical Dijkstra Algorithm for Node-constrained Shortest Path [J]. Journal of South China University of Technology (Natural Science Edition), 2017, 45(01): 66-73.
[11] Luyu Lan. Research on Urban Road Congestion Relief Method Based on Multiple Shortest Paths [C]. Shenzhen University, 2018.
[12] T. A. J. Nicholson. Finding the Shortest Route between Two Points in a Network. The Computer Journal, 1966, 9 (3), 275-280.