IMPROVE ON DIJKSTRA SHORTEST PATH ALGORITHM FOR HUGE DATA

Fuhao ZHANG*, Ageng QIU, Qingyuan LI

Chinese Academy of Surveying and Mapping, Beijing, China, 100039, Zhangfh@casm.ac.cn

KEY WORDS: GIS, Network, Network analysis, Shortest path analysis, Dijkstra

ABSTRACT:

This paper introduces the classical Dijkstra algorithm in detail, and illustrates the method of implementation of the algorithm and the disadvantages of the algorithm: the network nodes require square-class memory, so it is difficult to quantify the shortest path of the major nodes. At the same time, it describes the adjacent node algorithm which is an optimization algorithm based on Dijkstra algorithm. The algorithm makes full use of connection relation of arcs in the network topology information, and avoids the use of correlation matrix that contains substantial infinite value, making it more suitable analysis of the network for mass data. It is proved that the algorithm can save a lot of memory and is more suitable to the network with huge nodes.

1. INTRODUCTION

With the popularity of the computer and the development of the geographic information science, GIS has been increasingly extensive and in-depth applications for its powerful functions. As one of the most important functions, network analysis has played an important role in lots of fields, such as electric navigation, traffic tourism, urban planning and electricity, communications, and other various pipe network designs and so on. The key problem about network analysis is his shortest path analysis. The shortest path analysis not only refers to the shortest distance in general geographic sense, but also extends to other measurements, such as time, cost, and the capacity of the line. Correspondingly, the shortest path analysis is turned into the problem of the fastest path, the lowest cost and so on. With the map scale of the nationwide increased, such as the national map with the scale of 1:5 million(the total 20,000 pieces all round the nation, each ones including about three thousands nodes), the huge network analysis is necessary. The classical algorithm Dijkstra is the theoretical foundation for solving the problem about the shortest path. This paper presents a new algorithm based on the topological relation of the vector data to meet the network analysis for huge data in practice. First the paper introduces the Dijkstra algorithm in detail, and then proposes the network analysis for huge data.

2. DIJKSTRA ALGORITHM

2.1 The principle of Dijkstra algorithm

Hypotheses that D = (V,A,w) is non-negative weights network, V = (v1,v2,⋯,vn). Then the min D (vi,vj) ∈A satisfies the function:

$$u_1 = 0$$
$$u_j = \min(u_k + w_{kj}) \qquad (j=2,3,\cdots,n) \qquad (1)$$

The shortest path from vertex v1 to other vertex in D arranges from large to small as follows:

$$u_{i1} \le u_{i2} \le \ldots \le u_{in}$$

Here $il=1, u_{i1}=0$, then from equation (1) we can obtain:

$$uij = \min_{k \ne j}\{uik+wikij\}$$
$$= \min\{\min_{k<j}\{uik+wikij\}, \min_{k>j}\{uik+wikij\}\} \quad (j=2,3,\ldots,n)$$

If k>j, uik≥uij,and wikij≥0, then we can obtain:

$$uij \leqq uik+wikij$$

So

$$uij \leqq \min_{k>j}\{uik+wikij\}$$

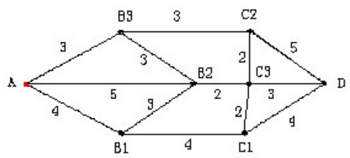
So

$$uij = \min_{k<j}\{uik+wikij\}$$

We can get the following equation easily:

$$ui1 = 0$$
$$uij = \min_{k<j}\{uik+wikij\}$$

In (ui1,ui2,⋯,uin), uij is the shortest length of (ui,uj), j=1,2,⋯,n。



$$\begin{Bmatrix} 0&1&1&1&0&0&0&0 \\ 1&0&1&0&1&0&0&0 \\ 1&1&0&1&0&0&1&0 \\ 1&0&1&0&0&1&0&0 \\ 0&1&0&0&0&0&1&1 \\ 0&0&0&1&0&0&1&1 \\ 0&0&1&0&1&1&0&1 \\ 0&0&0&0&1&1&1&0 \end{Bmatrix} \begin{matrix} A \\ B1 \\ B2 \\ B3 \\ C1 \\ C2 \\ C3 \\ D \end{matrix}$$

adjacent matrix

$$\begin{matrix} A & B1 & B2 & B3 & C1 & C2 & C3 & D \end{matrix}$$
$$\begin{Bmatrix} 0&1&1&1&\infty&\infty&\infty&\infty \\ 1&0&1&\infty&1&\infty&\infty&\infty \\ 1&1&0&1&\infty&\infty&1&\infty \\ 1&\infty&1&0&\infty&1&\infty&\infty \\ \infty&1&\infty&\infty&0&\infty&1&1 \\ \infty&\infty&\infty&1&\infty&0&1&1 \\ \infty&\infty&1&\infty&1&1&0&1 \\ \infty&\infty&\infty&\infty&1&1&1&0 \end{Bmatrix} \begin{matrix} A \\ B1 \\ B2 \\ B3 \\ C1 \\ C2 \\ C3 \\ D \end{matrix}$$

distancematrix

Figure 1. A network, its Adjacency matrix and its distance matrix

* Corresponding author. Zhangfh@casm.ac.cn

A simple network is shown in figure 1. The adjacency matrix and the distance matrix are obtained based on the relationship of distance and the vertex. It is easy to get the shortest distance from vertex A to vertex D based on the Dijkatra algorithm.

2.2 The disadvantages of the Dijkstra algorithm

In the algorithm and computer program, correlation matrix, adjacent matrix and distance matrix are used to compute the shortest path based on network matrix of Dijkstra. Many N*N arrays are defined to store graphical date and compute. N is refered to the number of the network nodes. When the number of the nodes is very large, it occupies a lot of CPU memory. For example, when the number of the nodes is 3000(about the nodes of one road layer with the scale of 1:50000), it needs 4*3000*3000 = 36000000 bytes—36 MB memory, and if the number is 6000(about the nodes of two road layer with the scale of 1:50000), it needs 144 MB memory. If we do not improve the Dijkstra algorithm, the algorithm is very difficult to apply in network analysis for huge data.

3. THE ADJACENT NODE ALGORITHM OF THE SHORTEST PATH

With the increase of the network spatial data and the network nodes, it is not practical to use Dijkstra algorithm directly. In order to process the shortest path analysis for huge data, we must optimize the classical Dijkstra algorithm. An improved method based on Dijkstra——adjacent nodes algorithm is illustrated here.

The algorithm based on the adjacent matrix or the correlation matrix has a lot of 0 or ∞ elements. This invalid element occupies a lot of CPU memory to decelerate the computing speed. The adjacent node algorithm is to minimize the 0 or ∞ elements in the matrix. In practice, although there are many nodes and lines in a network, the number of the lines and nodes related with the network nodes are few. If we only record the nodes and lines related with the network nodes, it can reduce many invalid 0 or ∞ elements in the matrix. But the number of the max correlation nodes and the max correlation lines is changing, we must get the number of the max adjacent nodes as rows or columns to construct matrix. So we can call the shortest path method based on the point－line relations matrix as adjacent nodes algorithm. The algorithm can reduce the o or ∞ elements to save the memory and improve the computing efficiency.

$$\begin{matrix}1\\2\\3\\4\\5\\6\\7\\8\end{matrix}\begin{Bmatrix}2&3&4&0\\1&3&5&0\\1&2&4&7\\1&3&6&0\\2&7&8&0\\4&7&8&0\\3&5&6&8\\5&6&7&0\end{Bmatrix}\qquad\begin{matrix}1\\2\\3\\4\\5\\6\\7\\8\end{matrix}\begin{Bmatrix}4&5&3&\infty\\4&3&4&\infty\\5&3&3&2\\3&3&3&\infty\\4&2&4&\infty\\3&2&5&\infty\\2&2&2&3\\4&5&3&\infty\end{Bmatrix}$$

adjacent matrix judgment matrix

Figure 2. the adjacent matrix and judgment matrix

The workflow of the proposed algorithm is as followed:

(1).According to the concept of the number of the max adjacent nodes, get the number of the max adjacent nodes m, which is equal to 4;

(2).Construct the adjacent matrix fJ. The nodes are as rows, the adjacent nodes are as columns. The number of the matrix's rows is equal to the number of true nodes in the network, and the number of the matrix's column is equal to the max size of the adjacent node in the network—m. The number of the line connected the node named i is as the NO.i row in the matrix. If the number of the adjacent node of the NO.i node is less than m, the number of the adjacent node is 0(See in figure 1 and figure 2).

(3).Construct the judgment matrix pJ. Compared with the adjacent matrix, the judgment matrix pJ is constructed by the number of the line of each element in the adjacent matrix instead of the number in the same position.

(4).The shortest path is computed between any two points based on the adjacent matrix fJ and the judgment matrix pJ.

And then, the shortest path between A and D is computed, the steps is as followed:

1. Initialize the temp label vector T, Ti=0, i=1, 2….8.

2. Find the value in the beginning row which is not equal to infinite in the judgment matrix, and initialize the distance vector D and the marking vector P:

D2=4,
P2=1,
D3=5
P3=1,
D4=3,
P4=1.

3. Compute the minimum in the distance vector fmin=3.

4. Valuate the corresponding marking vector T4=-1, and get the nodes i=4.

5. Find the value in i row is not equal to zero, according to function.

uj = min(uk+wkj) (j=2,3,…,n)

Compute the distance between the nodes relating with the node i and the beginning node in the loop, and judge whether the distance is less than the value in the distance vector, if it is true, the valuation distance vector D and the marking vector P are: For example the NO.6 node in the fourth row

u6=u4+u46=D4+U46=3+3=6

6<D6=∞

So the valuation distance vector D6 is 6, the marking vector P6 is 4.

Repeat the third step:

$$\begin{matrix}1\\2\\3\\4\\5\\6\\7\\8\end{matrix}\begin{Bmatrix}2&3&7&0\\1&3&6&0\\1&2&8&7\\1&3&8&0\\2&7&6&0\\4&7&4&0\\3&5&5&8\\5&6&4&0\end{Bmatrix}\qquad\begin{matrix}1\\2\\3\\4\\5\\6\\7\\8\end{matrix}\begin{Bmatrix}2&3&4&0\\1&3&5&0\\1&2&4&7\\1&3&6&0\\2&7&8&0\\4&7&8&0\\3&5&6&8\\5&6&7&0\end{Bmatrix}$$

adjacent matrix (1) adjacent matrix (2)

$$\begin{matrix}1\\2\\3\\4\\5\\6\\7\\8\end{matrix}\begin{Bmatrix}2&3&4&0\\1&3&5&0\\1&2&4&7\\1&3&6&0\\2&7&8&0\\4&7&8&0\\3&5&6&8\\5&6&7&0\end{Bmatrix}\qquad\begin{matrix}1\\2\\3\\4\\5\\6\\7\\8\end{matrix}\begin{Bmatrix}4&5&3&\infty\\4&3&4&\infty\\5&3&3&2\\3&3&3&\infty\\4&2&4&\infty\\3&2&5&\infty\\2&2&2&3\\4&5&3&\infty\end{Bmatrix}$$

adjacent matrix(3) judgment matrix (1)

$$\begin{matrix}1\\2\\3\\4\\5\\6\\7\\8\end{matrix}\begin{Bmatrix}4&5&3&\infty\\4&3&4&\infty\\5&3&3&2\\3&3&3&\infty\\4&2&4&\infty\\3&2&5&\infty\\2&2&2&3\\4&5&3&\infty\end{Bmatrix}\qquad\begin{matrix}1\\2\\3\\4\\5\\6\\7\\8\end{matrix}\begin{Bmatrix}4&5&3&\infty\\4&3&4&\infty\\5&3&3&2\\3&3&3&\infty\\4&2&4&\infty\\3&2&5&\infty\\2&2&2&3\\4&5&3&\infty\end{Bmatrix}$$

judgment matrix(2) judgment matrix(3)

Figure 3. compute the distance through the adjacent matrix and the judgment matrix

3. Compute the minimum value in the distance vectors fmin. If the marking vector value is equal to -1, do not include it, fmin=4.
4. Valuate the corresponding marking vector T2=-1, and get the nodes i=2.

5. Find the value in i row is not equal to zero, according to function.

$$u_j = \min(u_k+w_{kj}) \quad (j=2,3,\ldots,n)$$

Compute the distance between the nodes relating with the node i and the beginning node, and judge whether the distance is less than the value in the distance vector, if it is true, the distance vector D and the marking vector P are:

For example the NO.5 node in the second row

$$u_5=u_2+u_{25}=D_2+U_{25}=4+4=8$$

$$8<D_5=\infty$$

So the valuation distance vector D5 is 8, the marking vector P5 is 2.

Repeat the third step until the minimum value is equal to the NO.8 in the distance vectors; the value is the required shortest distance.

If the end node 8 is not computed, the minimum value is infinite, so the short path is not existed.

According to the marking vector P, we can easily trace the path. The above adjacent nodes algorithm avoids the correlation matrix with many ∞ elements, and meet the huge nodes data. According to 3000 nodes in the road layer of a standard vector map with 1:50000, if the distance between two points is across 10 maps, it contains 30000 nodes. If the max adjacent nodes is considered as 10, it totally needs 4*10*3000=120000 bytes—120K. If we magnify the 10 times than its size, it only needs 1.2 MB memory. The experiment shows that the algorithm can save lots of memory for the network with lots of nodes.

The proposed algorithm not only saves memory but also is effective. The result based on the proposed algorithm is shown in Figure 4. The number of the nodes is 12000. It is only need one second to finish the computation. The Dijkstra algorithm will need 6 seconds if the memory is enough. It is proved that the proposed method is in efficiency.

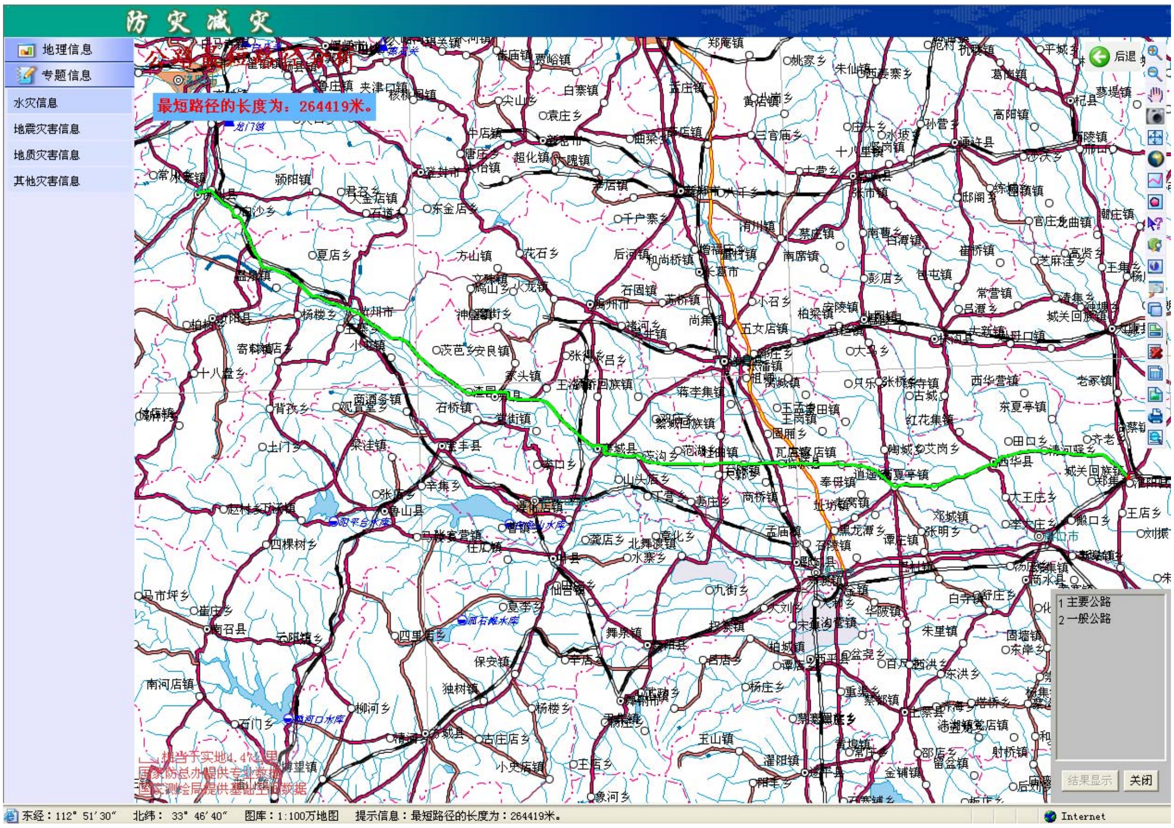


Figure 4. The shortest distance based on the proposed algorithm

4. DISCUSSION

This paper introduces the principle of Dijkstra algorithm, and then proposes an improved shortest path algorithm. The proposed algorithm makes use of the connections among the arcs in the network topology information to save memory to avoid the correlation matrix. The proposed algorithm is applied to the network with huge nodes to support the network analysis of the many maps after the pre-processing. The further improved algorithm can be met the network analysis about the setting of the limitation of the turning direction and the required blocking.

ACKNOWLEDGEMENTS

This research was funded by Key Laboratory of Geo informatics of State Bureau of Surveying and Mapping under grant B2534, and National High Technology Research and Development Program of China (863 Program) under grant No. 2007AA12Z215 and No. 2007AA12Z333.

REFERENCES

[1] Wu Qishi, The Application of Genetic Algorithm in GIS Network Analysis, *International Archives of Photogrammetry and Remo*, 2000, 33, 1184-1191.

[2] Benjamin Zhan F. Three Fastest Shortest Path Algorithms on Real Road Networks. *Journal of Geographic Information and Decision Analysis*, 1997, 1 (1): 69-82.

[3] Liu Yanliang, Wang Pengtao, Optimization model of complicated network and shortest path algorithm, *Journal of Tianjin University of Technology* 2006, 22(1): 42-44.

[4] Wang Jiechen, Zhang Wei, Mao Haicheng, Study of Graph Simplifying Method in GIS Network Analysis, *Acta Geodaetica et Cartographica Sinica*, 2001, 30(3): 263-268.

[5] Zeng Wen, Design of MAPGIS Pipeline Management Development Platform, *Journal of China University of Geosciences*, 2002, 27(3): 250-254.

[6] Liu Fangai, Liu Zhiyong, Qiao Xiangzhen, A Hierarchical Network HRN AND its routing Algorithms, *Chinese Journal of Computers*, 2002, 25(12): 1397-1441.

[7] Wang Jiechen, Yang Dezhi, Zhang Wei, An Improvement Algorithm of Shortest Route Analysis, *Journal of Institute of Surveying and Mapping*, 1999, 16(14): 282-285.

[8] Wang Jiechen, Mao Haicheng, Yang Dezhi, United Structure of Point-Arc for Network Graph and It s Application in GISs Shortest Path Searching, *Acta Geodaetica et Cartographica Sinica*, 2000, 29(1): 47-51.

[9] Li Lin, Variable Query Algebra and Shortest Path Analysis, *Acta Geodaetica et Cartographica Sinica*, 2000, 29(1): 47-51.