

Fast implementation of city optimal path

XiaoyongXu^a, JiquanZhou^b

Twenty-eighth Research Institute of China Electronic Technology Group Corporation, Nanjing,
Jiangsu province, China, 210018

^axyxeu@126.com, ^bjiqun51@163.com

Keywords: Optimal path; Time complexity; Dijkstra algorithm

Abstract. This paper gives a detailed analysis of the existing city shortest path algorithm[1], aiming at the high time complexity and can not represent the real situation of the problem of the existing city shortest path selection algorithm, and bases on the idea of Dijkstra algorithm, puts forward a fast algorithm of optimal path of the city. This algorithm can be considered to simulate the real situation may in the road grade, the degree of congestion, traffic light quantity and other factors, at the same time can be optimized in the time complexity of the algorithm, to achieve the shortest path fast search. The experimental results show that this method can simulate the real situation in as far as possible under the condition, reduces the time complexity of the algorithm, and get feasible result.

1 Introduction

Along with our country urbanization pace faster and faster, the city scale expands unceasingly in our country, the urban traffic flow increase, which brought great pressure to urban traffic. At present, China's major cities are faced with traffic problems, of which the problem of traffic safety is particularly serious. So, this requires our social functions in an accident to make rapid response, rushed to the scene of the accident rescue in time, and made the life and property losses to a minimum. Then rescue vehicles travel the shortest path will play a key role in the rescue work. The shortest path not only refers to the shortest distance in general geographical sense, but also can be extended to other metrics, such as time, cost, line capacity, etc... Accordingly, the shortest path problem has become the fastest path problem and the minimum cost problem, etc... The shortest path problem often used in vehicle navigation system and a variety of emergency systems in the actual (such as the 110 alarm system, 119 fire alarm system and emergency medical service system). These systems are generally required to calculate the best route to the site of the accident time be within 1s ~ 3s. In the process of driving vehicles in front of the line needs to be real-time calculated, this determines the shortest path problem should be high efficiency. In fact, whether it is the shortest distance, the shortest time or the lowest cost, their core algorithm is the shortest path algorithm. The classical shortest path algorithm - Dijkstra algorithm is the theoretical basis for most systems to solve the shortest path problem, but the different systems uses different implementation methods of Dijkstra algorithm.

According to statistics, there are about 17 kinds of algorithms for the shortest path. F.B enjaminZhan et al., 15 of those kind of algorithms are tested, and the result shows that there are three algorithms is better, they respectively are: TQQ, DKA and DKD. The TQQ algorithm is based on the theory of graph growth. It is more suitable for calculating the shortest distance between single source and all other points. The other two algorithms are based on the Dijkstra algorithm, more suitable for the calculation of the shortest path problem between two points. Generally speaking, the data structure and implementation method of these algorithms are restricted by the development level of computer hardware at that time. Space storage is very tricky, usually in exchange for saving space at the expense of the appropriate time efficiency. At present, the space storage is not the main problem to be considered, so it is necessary to consider and improve the existing algorithms, which can improve the efficiency of the shortest path algorithm.

Traditional Dijkstra algorithm for urban road time complexity $O(n^2)$ of the shortest path, although is more time to get the accurate optimal solutions. Then there is a series of improved

algorithms, which can reduce the time complexity of the algorithm to $O(n)$. Although these algorithms obtained solutions are not necessarily exact solutions, but in some cases these solutions are acceptable. How to get an acceptable feasible solution, which is the most close to the exact solution of feasible solution. This is the topic of this article.

2 Key technologies and theoretical basis

2.1 Calculation of the shortest path

Urban road network model is $G = (V, E)$. Among them, the V for site after comprehensive set of node E for the distance between two nodes is segmental arc distance, it is a nonnegative real number.

$$E = \{dab \mid dab = F(X1, X2, X3 \dots)\} \quad (1)$$

Eq.1 X1: simple distance factor, said the road geometric distance; X2 for road level factor; X3 congestion factor.

X2, X3, said the road level and the people, the influence of traffic and other factors on the road traffic capacity, you can use the impedance values to represent the influence degree. Impedance value of the size of the need to classify the whole level of road traffic zone and congestion, and then in turn for each kind of impedance values. In this way, the magnitude of the impedance value represents the influence of the road level and congestion level on the road traffic capacity. Although this effect is only qualitative, and can't accurately represent the size of the impact, but it has a horizontal comparison.

2.2 Dijkstra shortest path algorithm.

The basic idea of Dijkstra algorithm[2, 3] is that if each point has a label (d_j, p_j) , d_j is the length of the shortest path from the origin point s to point j (from a point to its length is 0, the distance between two points which are not communicated with each other for), p_j is a point before j in the shortest path from s to j . From the origin point s to j shortest path algorithm is the basic process is:

(1) Initialization, the starting point is set to: ① $d_s = 0, p_s$ is empty; ② all other points: $d_t = \infty$, finding the shortest path on different points, p_t is different; ③ marked the origin point s , denoted by $k = s$, All the other point is set to the unlabeled.

(2) Check the distance from all tagged point k to the direct connection of unlabeled points j , and set up

$$d_j = \min \{d_j, d_k + l_{k,j}\} \quad (2)$$

Eq.2: $l_{k,j}$ is the direct connection distance from k to j .

(3) Select the next point, from all the unlabeled nodes, choose the smallest i value in d_j .

$$d_i = \min \{d_j, \text{all the unlabeled } j\} \quad (3)$$

Corresponding points i , in the shortest path and set for the calibration.

(4) Find a point before i . Find u from the marked point, which direct connection point i , and set $i = u$

(5) Gauge point i . If all points have been marked, the algorithm exit, or $k=i$ Eq.2, to continue.

From the above, more vertexes, more cycles, the computing time will grow exponentially. Graph traversal Dijkstra algorithm used is number of vertices as the base for the secondary loop[4], and each cycle times are the number of nodes n . It became to the execution process of s for the node of a tree, with the algorithm of further extending in all directions, and perform until reach the destination. The time complexity of algorithm implementation for $O(n^2)$ [5], it can find s to all nodes of the shortest path from the source point. Obviously, in the end, some branches is useless to obtain the shortest path. Whether delete these branches, thereby reducing the time of computing the shortest path, this is the reason of the improved algorithm is presented.

2.3 Improved algorithm

Ditto, set s and j is any given nodes in the network, to find the optimal path of s to j, which path is made up of edges and nodes. Obviously segment is the shortest path between two points without considering the real. Namely the path which connection s, j is the shortest path. It rarely appears in the actual case, but you can know the closer of the line is the better choice. That is to say, the least Angle of this path. You can use the GIS technology to find a feasible solution under the goal.

Algorithm thought is set off from s in search of the adjacent nodes, and put them together. Assuming find a point s_1 , make it with s, j of the angle of the smallest point, and start from s_1 to find the next point, which node with s_1, j of the smallest Angle. And this recursive down, when the search to the end of j is out of the loop. Because the subject is in the study of urban road, based on the rationality of the urban design of the path (every site is accessible), and always can find such a path.

2.4 Algorithm analysis and improvement method

The last chapter of the article has been put forward the basic solution, the following analysis of the rationality, and superiority of the algorithm. And puts forward the further optimization method. After further analysis, it was found that there are some problems in the algorithm above. As shown in figure:

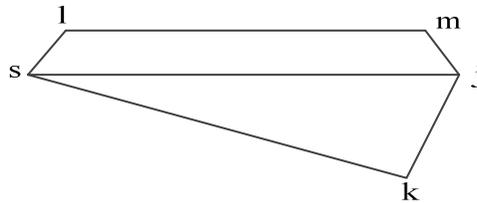


Fig. 1 analysis of model algorithm

If search according to the above method that it will apparently get the answer of $s \rightarrow k \rightarrow j$. But this answer is not as good as $s \rightarrow l \rightarrow m \rightarrow j$ obviously, after analysis found that the path is too long from s to k. Then we need to consider the influence of the size of the included Angle and the length of the path on the path selection. I put forward the following solutions.

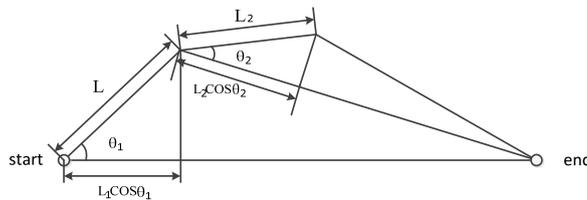


Fig. 2 link model

From the start node to start looking for neighboring node in city road on the electronic map, then according to the size of V_t select the location of the next node of the optimal path; With V_t calculated according to Eq. 4.

$$V_t = L \cos \theta V_i \omega / (L + C V_i \omega + a N V_i \omega) \quad (4)$$

Eq. 4 L represent the length between the next node and previous determined node, θ represent the Angel between the next node, previous determined node and the end of the link, V_t represent the speed of the road i (density tends to zero, the average speed of vehicles travel around the checkless space), $V_i \omega$ represents average speed in the i stretch of road space, its value is expressed as Q_i / k_i (km/h), expressed as the average flow on the road i (truck /per hour), Q_i are expressed as the average density of on the road i (truck per km), N represent the number of traffic lights, a represent the spending time in unit of traffic light, C is fixed time, (fixed the influence on the running time, what because the turning vehicle ,traffic light caused by the vehicle deceleration, or some other).

3 Project objectives

This project was developed in the area of Shahu Suzhou ecological park, the selected area with Coverage to establish the ArcGIS data model. GIS is based on the geographic data model, to realize the storage, organization, processing, and the representation of geographical data in the computer. The main types of Coverage elements are points, arcs (lines), polygons and nodes. On the basis of the establishment of the Coverage data model, and the data obtained, then through the field trips, adding other no basic information (such as traffic lights, road grade, etc.), and form of a relational database. And form two data tables, node attribute table, save the selected area of the data of each node, as shown in Table 1, and a section of the attribute table to save the information of the various sections, as shown in Table 2.

Node attribute	database field
Node code	NodeID
Node name	NodeName
Nodelongitude	NodeLon
Node latitude	NodeLat

Section attribute	Database field
Section code	RoadID
Section name	RoadName
Traffic light quantity	LightNum
Section level	RoadGra
Section length	RoadLeg
The degree ofcongestion	RoadCw
Road starting	RoadF
At the end of section	RoadE

Develop it based on the statistics, which is to give the best way immediately and can display it on the map in case of importing the starting point as well as the terminal point of.

4 Design

4.1 Implementation plan

Collecting data in the selected area at first, and using ArcGIS to build coverage data model, then modified after the on-the-spot investigation, eventually be able to use the database, and then determine the solving model and method, is solved, and the model is verified. As shown in Fig. 3.

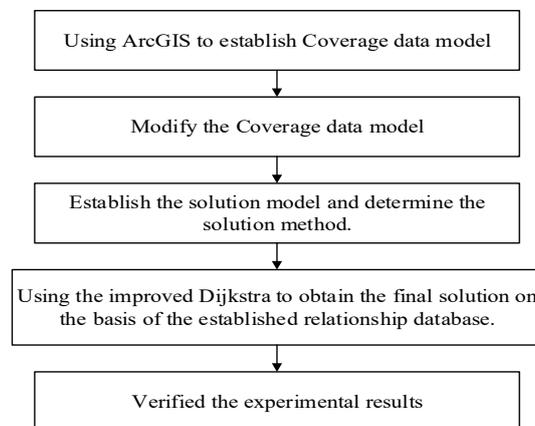


Fig. 3 steps

4.2 Solution

Let s and j are two nodes in any given network, steps to find the optimal path algorithm for s to j :

The first step: each parameter initialization, determine the level of road vehicle speed table (Table 1)

The second step: search from the i node, and according to the size of the V_t to search the next node j (j has the maximum V_t near the node of i)

The third step: if the j is the end of the search, exit the algorithms, if not returns to second step. If the j node is not neighboring nodes and not the end, rounding the path and returns the second step to search.

The fourth step: the final node of the search is the end, the end result. As shown in Fig. 4:

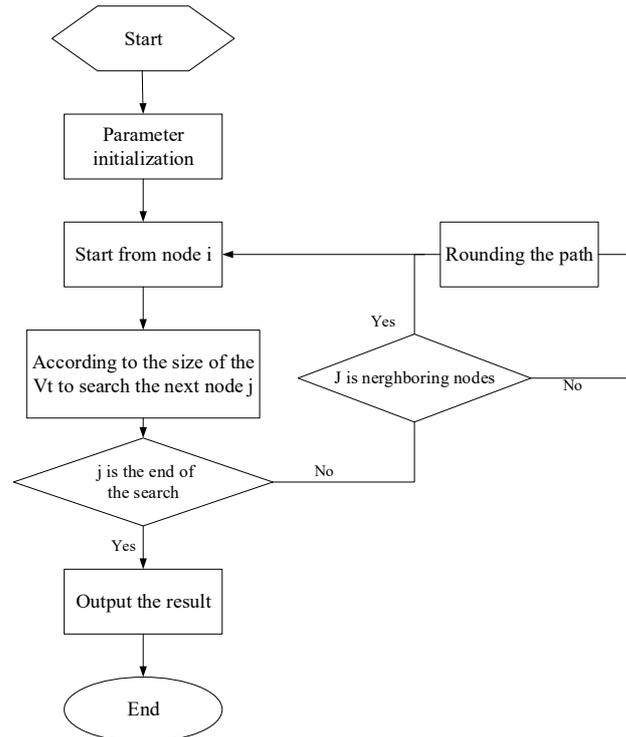


Fig. 4 solving process diagram

5 Implementation and testing

The system consists of the module for line searching as well as the module for routine display. There are 161 nodes in all. The system can correctly display the information of roadway if there is nothing wrong with data-in. For example, right routine should be displayed when Node 2 and Node 145 are input. When we are choosing different projects, routines search for are able to be correctly displayed and shown on the map as we click on the display-routine-button. As is shown in Picture 5, Picture 6, here are the display of the routine when Project 1 and Project 2 are chosen.

