

AN OPTIMAL SPATIO-TEMPERAL PATH ALGORITHM FOR URBAN EMERGENCY RESCUE

Min Cai^{a,*}, Yuejin Deng^a, Zhongri Tang^a

^a Wuhan University, LIESMARS, 430079 Wuhan, Hubei, China –standby211@qq.com, whudyj2010@gmail.com

KEY WORDS: Optimal path, Time influence model, Emergency rescue

ABSTRACT:

As the development of the Geospatial Information System (GIS), the application of GIS has spread a variety of realms such as urban plan, resource management, transportation, environmental protection, surveying, and mapping. Especially the GIS-based path plan algorithm is widely applied for the rescue of emergency, such as earthquakes, forest fires, urban accidents, etc. In an emergency system, there are three essential keys to shorten emergency response time. Firstly, there should get real-time position of patrol cars. Secondly, it should obtain emergency resources timely. Lastly, there should be a real-time spatial search method which is based GIS/GPS emergency response system to get the shortest path for best emergency response force. There are many algorithms for optimal path computing. Most of these algorithms can give the shortest path results. However, since the shortest path is not surely the best path with least time-consuming, it is necessary to consider some factors such as the number of lanes, traffic volume, road length, road conditions, driving rules in the optimal path computing. In this paper, we provide an improved quick Dijkstra algorithm regarding these factors, instead of taking the shortest distance as target function, the least time is primary taken into account and the time influence model is used in this method. A simulated case study for urban emergency rescue using high-precision geospatial data and all related information such as lanes, traffic volume, length, conditions, and driving rules shows this approach feasible.

1. INTRODUCTION

Since urban emergency incidents have the characteristics of unpredictability, continuity, calamity, denseness, diffusivity and sociability, incidents occurred in different time, places and immediate environments vary. Besides, the state of the city's fire fighting power and the real-time traffic conditions are in a dynamic changing position. These factors have greatly increased the difficulty of rescue. Once the rescue is not timely given, it will bring huge casualties and property losses. Therefore, an optimal spatio-temporal path algorithm based on GIS for urban emergency rescue has great significance, which can be used to assist emergency rescue, reduce losses and control the disaster.

Many scholars had made researches on the emergency rescue systems using GIS technology as a tool to the assessment and rescue of forest fires, urban fires and so on. Tsai Yichang, et al. (2002) proposed a prototype of real time Emergency Response System (ERS) based on GIS/GPS. There are three essential keys in an emergency system to shorten emergency response time. Firstly, there should get real-time position of patrol cars. Secondly, it should obtain emergency resources timely. Lastly, there should be a real-time spatial search method which is based GIS/GPS emergency response system to get the shortest path for best emergency response force. Kurt Mehlhorn and Guido Schaefer (2001) proposed the popular A*Heuristic algorithm applied in transportation network path optimization. WangJing Cun, et al. (2007) applied Graph Theory and other related knowledge to propose the optimal path for emergency rescue computational modelling based on Floyd Algorithm, which combined with transportation weighted graph. Skriver AJV and Andersen KA (2000) applied the Dijkstra shortest path algorithm to the urban intelligent fire fighting management

system based on GIS. Agustina B. and Coello CA (updated 2010) give a list of more than 4800 references on evolutionary multi-objective optimization.

At present, the shortest path algorithm mainly includes Floyd algorithm, Ant Colony algorithm, A* algorithm and Dijkstra algorithm, etc. (V. Vassilevska, etc., 2006). The Floyd algorithm and the related extended algorithms were designed to solve the shortest path between any two vertexes by calculating the weight matrix of graph. These algorithms' time complexity is $O(n^3)$, allowing extra edges with negative weight, but not with negative weight circles. The Ant Colony Algorithm is a distributed algorithm with adaptive features and a randomly searching algorithm, which does not require a large number of probability calculations and the creation of complex mathematical models, easy to implementation. The A* algorithm is a multi-source algorithm to calculate the shortest path between points (J. C. N. Cl ímaco and M. M. B. Pascoal, 2007). The Dijkstra algorithm is a single-source algorithm, computing the shortest path from a select point to other points. In order to guarantee the shortest distance from the starting point to the current node, the Dijkstra algorithm requires searching all visited nodes, including the anti-direction searching. In this paper, the improved Dijkstra algorithm is used by calculating the sum of the path weight from the starting point to the current node as the greedy selection strategy.

Network topology structure is the basis for dealing with path analysis. WangJing Cun, et al. (2007) presented an approach to the network topology structure' extraction and building in the experimental system for shortest path analysis. The A * Algorithm define data structure for the organization of road network topology.

* LIESMARS, Wuhan University, 430079 Wuhan, Hubei, China. Email:standby211@qq.com.

Nowadays, with the development of GIS applied in urban emergency rescue, there still exist some problems in optimal path planning. Firstly, the majority of the urban emergency rescue systems based on GIS has implemented the shortest path planning algorithm, but does not consider how to implement the shortest path planning in the condition of real-time traffic. Secondly, there is lack of overall planning for the distributed rescue power. Thirdly, there are some problems in the implement of algorithm efficiency. Lastly, there exist some irrational aspects in comprehensive analysis of the various road factors affected the rescue time. This paper focuses on the optimal path planning and algorithm efficiency. Also, we simulate the actual time consuming in paths by utilizing time influence model which provides scientific decision-making to support rapid emergency rescue.

2. TIME INFLUENCE MODEL FOR PATH PLANNING

2.1 Urban Emergency Rescue System

The data flow diagram of the urban emergency rescue system is demonstrated in Figure 1.

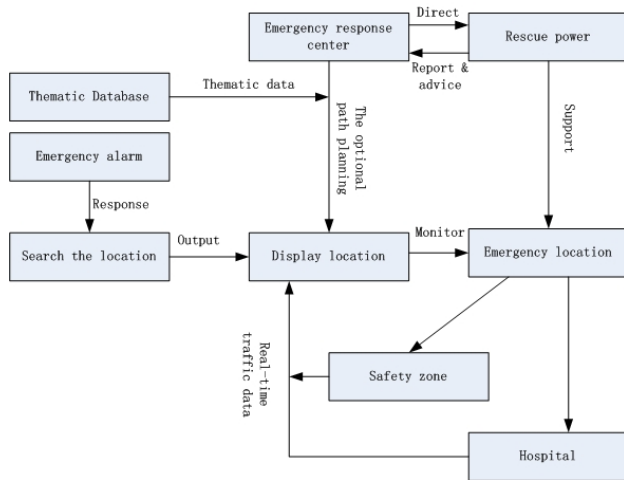


Figure 1. Data flow of the urban emergency rescue system

2.2 Topological Data Structure for Urban Road Network

Network topology structure describes the network connectivity relationships of the points, lines and planes. The classic storage structure for road network topology data mainly includes adjacency matrix and adjacency list, etc. (X. Gandibleux, F. Beugnies and S. Randriamasy, 2006). When we use adjacency matrix to store a large number of road network with sparse graphs, it would cause great waste of space storage although the relationship of the nodes' in-out degree is obvious. Compared to adjacency matrix, adjacency list significant saves storage space. At the same time, it requires storing not only the data of road network of nodes but also the linking information such as the chain domain data; also the relationship of the in-out degree is not obvious. Therefore, we define a new topological data structure for road network data identified with nodes, which applies to the Dijkstra algorithm. The topological data structure is shown as follow:

Struct Node

{

```

int x;
int y;
int ID;
bool blsOpen;
bool blsClose;
int ParentIndex;
float H;
float G;
float F;
int NeighborCount;
std::vector<int>Neighbor;
std::vector<float>NeighborDistance;
};

```

Where x, y is the node coordinate after transformation, ID is the node's ID, $blsOpen$ and $blsClose$ is used to identify the nodes in the Open table or Close table or neither, $ParentIndex$ is the current node's parent node index, H, G, F is the variables of the objective function, $F=G+H$, $NeighborCount$ is the count of nodes adjacent to the current node, $Neighbor$ is the index of nodes adjacent to the current node, $NeighborDistance$ is the weight corresponding with the $Neighbor$ from the current node to the adjacent nodes. $NeighborDistance$ stored the same order as $Neighbor$.

The representation of the road network means how the road networks are represented and stored in the computer (Hershberger, J.,Suri,S, 1999). It is the basis of calculation method of the optimal path planning. The whole road network is displayed as a graph that consists of all nodes and arcs. Therefore, the optimization of topological data structure for urban road networks has greatly reduced the stored space and has greater efficiency than the not optimization stored ways, which helps to significantly reduce the running time, costing only about 70% of the time of the previous way (Pettie, S., Ramachandran, V., and Srinath,S,2002).

2.3 Time Influence Model

According to the knowledge of emergency rescue, we could get the unreliable search results if we just search the shortest path by its length. The better realistic path planning could be found if the number of lanes, traffic volume, road condition and driving rules and other factors impacting on the actual driving time are under consideration. According to the thought of adaptive control and all kinds of factors, it is necessary to model and analyze of various factors in the optimal path planning. In our approach, we take the shortest time as the objective function in the optimal path planning with four factors: number of lanes, traffic volume, road condition and driving rules.

$$K = K(L, N, C, R) = (K_{(C)} \times K_{(L)}) / (K_{(N)} \times K_{(R)})$$

Where L represents for the coefficient of the number of lanes, N is the coefficient of traffic volume obtained and updated from real-time transportation database, C represents for the

coefficient of road conditions, R for the coefficient of driving rules, and K_0 for the coefficient of the weight determined by experts for each factor.

In this time influence model, we should obtain the related real-time information and update the database immediately to extract the need information, which increased the algorithm's complexity in some respects. However, we have established the adaptive mathematical model to dynamically determine the weight of each road, and the weight was the essential information for searching the optimal path. In the next part of this paper, we will introduce how to get the different coefficients in the above model.

2.4 The Number of Lanes

According to urban road design technical standards in china, roads are classified into four grades, including expressway, arterial road, secondary road, and branch road. Different grades correspond to different road capacity (see Table1).

Road grading	Expressway	arterial	Secondary road	Branch road
Road capacity(pcu/h)	1600	1400	800	400

Table 1.Road capacity

So we define the value of the number of lanes as:

$$K_{(L)} = \text{grade X road capacity} / \text{branch road capacity.}$$

2.5 Traffic Volume

According to the habits of living and working in china, there are four traffic peaks every-day. They are two outgoings in the morning and afternoon and two returns at noon and night. We define the time it takes as t_1, t_2, t_3, t_4 . The percentage of the whole-day traffic is n_1, n_2, n_3, n_4 and the formula $N_1 = (n_1 + n_2 + n_3 + n_4) \div (t_1 + t_2 + t_3 + t_4)$ is the percentage of the which average traffic volume each hour during traffic peak to whole-day traffic volume. And the formula

$$N_2 = [(1 - n_1) + (1 - n_2) + (1 - n_3) + (1 - n_4)] /$$

$(24 - t_1 - t_2 - t_3 - t_4)$ is the percentage of which average traffic volume each hour during not traffic peak to all-day volume. So we get:

$$K_{(N)} = N_1 / N_2.$$

2.6 Driving Rules

Using the Dijkstra algorithm, the shortest path may be forbidden, because the flow of vehicles goes one way only on one-way traffic. However, the one-way traffic could increase traffic capacity under the same road grade. According to traffic and transportation handbook of the United States, the road capacity could increase by 20% to 50% when the two-way

traffic is changed into one-way traffic. So we could get the weight coefficients of road (see Table 2).

One-way or two-way	One-way	Two-way	Forbidden
Weights coefficient	0.5	1	∞

Table 2. The weight coefficient of driving rules

2.7 Road Conditions

Different road conditions can impact on the driving, so we must take the road conditions into considerations. Road friction coefficients are the friction ability between tire and road; it relies on the road conditions. The drivers of rescue vehicles could drive safely according to the road conditions. So the road friction coefficients could describe quantitatively the road capacity in order to get the value of weights coefficient. We could get the formula of the road volume friction coefficient:

$$\mu_{xp} = 0.92 \times 0.1304e^{\sigma} + 0.002e^{\sigma} (64 - v) - 0.0426\sqrt{\mu}$$

Where σ as the coefficient of characterization of friction impact factor of the road conditions (see Table 3.),

v as the coefficient of speed ,

μ as the coefficient of carload , $\mu = F_X / F_Y$, F_X as coefficient of the actual carload of the tire, F_Y as coefficient of the standard carload,

μ_{xpb} as the coefficient of the standard volume friction coefficient when $\sigma = 0, v = 0, F_X = 0$,

So we get:

$$K_{(C)} = \mu_{XPB} / \mu_{XP}$$

Road condition	Dry	Wet	Incompact	Compaction	Ice-dry	Puddle
σ	0	0.13	0.6	0.75	1.0	1.2

Table 3. The common value of σ

3. THE OPTIMIZATION OF THE DIJKSTRA ALGORITHM

The Dijkstra algorithm is a classical algorithm for solving the optimal path. By using the Dijkstra algorithm, we search the shortest path according to road weights increasing (Wagner, D., Willhalm, T., and Zaroliagis, C., 2005). This method is effective for one-node to all-nodes or all-nodes to one-node. But by searching the shortest path between two nodes, the method is inefficient and $O(n^2)$ is very high because we have to search too many unnecessary nodes and lines. So we use the function

to solve the problems mentioned above. The function includes two parts. One is beeline distance is the shortest between two points, namely related information, the other is dynamic road weights (Pinto L, Pascoal M., 2009.). According to the beeline distance between two nodes is the shortest; the function limitation strategy is put forward to narrow search area, reducing the number of search nodes and the time complexity in order to accelerate the search speed. The path length is defined to the sum of the weight of all edges. And the specific meaning of the path length depends on the edges' weights representing the significance (Aleksandrov,L., Maheshwari,A., Sack,J.-R., 2003). In this paper, we set the weight of the edge as the actual passage of time, where T equalled that the road length (S) was divided by the result of vehicle limited speed (V) multiplied by the time influence factors (K). The basic idea of the algorithm is as follow: set S as the vertex sets with shortest distance having fixed, named in red vertexes set; set V-S as the vertex sets with shortest distance not having fixed, named in blue vertexes set. So the distance is defined as:

$$Di = \sqrt{(X_x - X_p)^2 + (Y_x - Y_p)^2}$$

Where x represents for the temp node, and p for the target node.

The process of the algorithm is described below.

(1) Initialization

Only the beginning point S' shortest distance was already known: SD (s) equalled zero. So the set of red vertexes S equalled {s} and the set of blue vertexes was empty.

(2) Calculating each vertex's shortest path (Di) toward the road length's increment direction.

(3) Choosing a blue vertex with minimum in the shortest path from the current set of blue vertexes to expand the red vertexes set.

(4) Repeating the second step when the blue dots' shortest path was remaining endless in the blue vertexes set or all of the vertexes have expanded to red vertexes set. Finally, the shortest paths from S to other vertexes have gotten.

4. A CASE STUDY

In this paper, we use the data of road network in Wuhan city and the data from Emergency Command Centres to simulation analysis. In this case study, we use ArcSDE to access and manage the spatial data in Oracle and use ArcObjects to visualize the planning results as shown in Fig. 2.

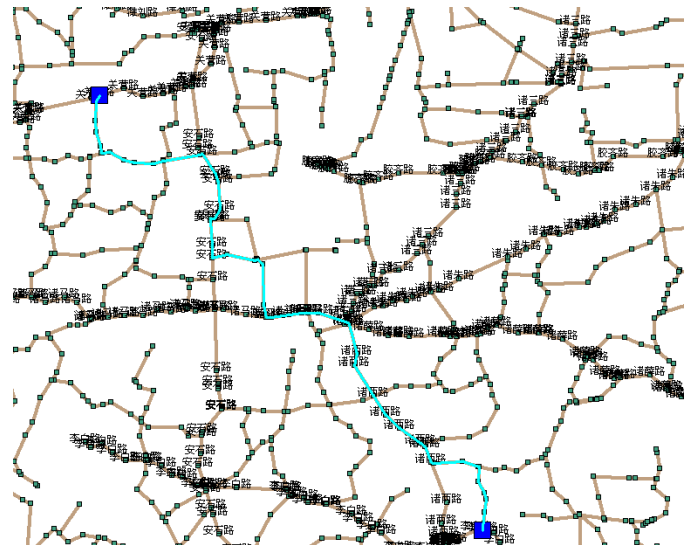


Figure2. A searching the optimal path from simulation

5. CONCLUSIONS

This paper proposes a fast algorithm for the optimal path planning based on GIS. On the basis of UGIS, the optimal algorithm suppose the shortest time-consumed from the location of each rescue unit to the location of the incident as an objective function, designed to provide scientific support for fast urban emergency rescue.

The Dijkstra Algorithm's time complexity is $O(n^2)$. Utilizing Node-based data structure, there stored the number of nodes adjacent to the current node, point to identify and the node to adjacent node weight in the data structure. There introduces the queue corresponding to the related operations such as insert, sort, delete, insert and delete, of which time complexity was $O(1)$ in the optimization of algorithm. However, the time complexity is $O(n \lg n)$ with fast sorting algorithm. Therefore, the whole algorithm's time complexity is reduced to $O(n^{3/2})$ through these methods. When n is large enough, the algorithm's time complexity is greatly lowered.

References

Agustina Buccella, Coello CA, 2010. List of References on Evolutionary Multi-objective Optimization. <http://www.lania.mx/~ccoello/EMOO/EMOObib.html> (accessed 13 Sept., 2010).

Aleksandrov,L., Maheshwari,A., Sack,J.-R., 2003. An Improved Approximation Algorithm for Computing Geometric Shortest Paths. In: Proc. of the 14th Internet. Sympos.Fundam. Computer. Theory (FCT'03).Lecture Notes in Computer Science, vol. 2751, pp. 246–257.

Hershberger,J.,Suri,S., 1999. An Optimal Algorithm for Euclidean Shortest Paths in the Plane. SIAM J. Computer. 28 (6), pp.2215-2256.

J. C. N. Cl imaco and M. M. B. Pascoal, 2007. "Finding non-dominated shortest pairs of disjoint simple paths", Technical Report, no. 3, INESC-Coimbra, pp.199-208.

Kurt Mehlhorn, Guido Schaefer, 2001. *A Heuristic for Dijkstra Algorithm with Many Targets and Its Use in Weighted Matching Algorithms. The 9th Annual European Symposium on Algorithms - ESA, Aarhus, pp.242-253.*

Pettie, S., Ramachandran, V., and Srinath,S., 2002. *Experimental Evaluation of a New Shortest Path Algorithm. UTCS Technical Report TR0137, In Proceedings of the 4th Workshop on Algorithm Engineering and Experiments (ALENEX02), Lecture Notes in Computer Science, pp.105-120.*

Pinto L, Pascoal M., 2009. *Enhanced Algorithms for Tricriteria Shortest Path Problems with Two Bottleneck Objective Functions, Technical Report No. 3, INESC-Coimbra, pp.1774-1779.*

Skriver AJV, Andersen KA, 2000. *A Label Correcting Approach for Solving Bicriterin Shortest Path Problems. Technical report, Department of Operations Research, University of Aarhus, Denmark. Computers and Operations Research, 27(6), pp. 507-524.*

V. Vassilevska, R. Williams, and R. Yuster, 2006. *Finding the Smallest H-subgraph in Real Weighted Graphs and Related Problems. In LNCS ICALP 33rd 4051, pp.262--273.*

WangJing Cun,ZhangXiao Tong, 2007. *A Heuristic Optimization Path-finding Algorithm Based on Dijkstra Algorithm. Journal of University of Science and Technology Beijing, 29(3), pp.346-350.*

Wagner, D., Willhalm,T., and Zaroliagis, C., 2005. *Geometric Containers for Efficient Shortest Path Computation, Journal of Experimental Algorithmics, 10(1), pp.15-47.*

X. Gandibleux, F. Beugnies and S. Randriamasy, 2006. *Multi-objective Shortest Path Problems with a Maxmin Cost Function, 4OR – Quarterly Journal of the Belgian, French and Italian Operations Research Societies (4), pp. 47–59.*

Yichang (James) Tsai, Zhaohua Wang, and Chien-Tai Yang, 2002. *A Prototype Real-Time GPS/GIS-Based Emergency Response System for Locating and Dispatching Moving Patrol Vehicles with the Beat-Based Shortest Distance Search. Proceedings of the Third International Conference on Traffic and Transportation Studies, pp. 669-678.*

Acknowledgements

This paper is supported by the project named Optimal Path Planning Algorithm for Urban Emergency System Based on GIS, which was sponsored by the Chinese Academy of Surveying and Mapping (Project No. 2009-04). The authors also appreciate the softwares from Wuda Geoinformatics Co., Ltd. in our experiment.