Weight optimization of transportation routing planning Dijkstra algorithm

Jun Yuan, Peilin Zhang*, Liqun Ding

Transportation planning and management, Wuhan University of Technology, Wuhan, 430063, Hubei, China

Received 1 June 2014, www.cmnt.lv

Abstract

As to standard Dijkstra algorithm for planning transportation routing still exists low computing speed, inaccuracy issues, this paper proposes a Dijkstra algorithm which is based on data constraints and path impacts optimization of factors, firstly, pre-process all directed-node pairs in routing traffic network, and then in Dijkstra way for reverse expanding boundary nodes, which limits its path boundary, then mark the optimal node on path nodes and take "target block" as a unit for storage, so as to speed up algorithm's running speed, at last, in order to improve the accuracy of traditional algorithm's routing planning, add routing impact factors to the traditional algorithm, and weight optimize it. Simulation results show that, compared with the standard Dijkstra algorithm, the Dijkstra algorithm proposed in this paper is based on data constraints and routing impacts optimization of factors, its speed of operation is greatly shortened, routing planning accuracy is greatly increased, and it has an good robustness.

Keywords: Dijkstra Algorithm, Transportation Routing Planning, Data Constraint, Reverse Expand, Routing Effected Factor, Weight Optimization

1 Introduction

With rapid development of economy, energy and environmental issues have become a constraining bottleneck of China's rapid economic development, energy conservation and emission reduction and low-carbon development will be the inevitable choice for China's future development. Intelligent transportation system makes full use of transportation infrastructures, improves service quality, and enables efficiently use transportation facilities and energy, thus obtains huge social and economic benefits [1]. Vehicle navigation system is an important system that urgently demanded, and widely used in intelligent transportation systems, it is one of development directions in whole modernized highway traffic system. Optimal routing planning is a critical issue in-car navigation system [2]. In the case of departure and destination are determined, according to certain strategies, such as time-saving or shortest path, etc., and topology information in electronic maps to help vehicle drivers or dispatchers accurately to select an optimal driving path in current time, and will display on computer screen's electronic map [3].

Static map searching algorithm is usually divided into blind searching algorithm and heuristic searching algorithm. The most classic blind searching algorithm is Dijkstra algorithm, it is very suitable in a weighted directed graph to obtain the shortest routing, but due to Dijkstra algorithm' large searching range, the algorithm efficiency is relatively low, so it has been greatly restricted in practical applications [4]. With the improvement and development of classical graph theory and computer technology, innovation in data structures and algorithms aspects promotes the emerging of new shortest routing algorithm. Bellman proposed a function equation technology used in dynamic programming, combined with policy space's approximate concept to obtain by combined a successive approximation method, which can solve manually or machine problems in reality [5]. Pape and others proposed Graph-Growth algorithm for the relative efficiency problem, it proved that using different data structures and implementation technique in the graph theory is effective [6].

Glover proposed a hybrid algorithm to solve routing planning problem, this algorithm has the feature of label setting and label correcting, and it has been proved by experiment that it possess features to overpass two algorithms, it's called THRESH [7]. Hart.et al and other scholars proposed A * algorithm, A * algorithm is a heuristic search algorithm, which makes use of finding heuristic factors related to issues to speed up the searching process, thereby reduces the time for searching graphs [8]. Luo Guoqing used the ant colony algorithm realized a dynamic optimal routing searching, and compared the simulation results of ant colony algorithm and Dijkstra algorithm, the authors adopted BPR function model introduced by US Highway Administration to calculate dynamic time, then took one day hour traffic flow as the road traffic flow [9]. Chen Yan used the ant colony algorithm as the optimal routing selection algorithm and designed the corresponding routing modules; took road distance as road weights, and the shortest routing distance as the target [10]. Jing Ling used a genetic algorithm to solve the optimal routing, and took path length as weights value [11]. Wang Rongyan also adopted genetic algorithms as the optimal routing

* ***Corresponding author's*** e-mail: yuanjun623@126.com

selection algorithm, what's different is that he constructed a dynamic path travel time function model to obtain road travel time, and took it as the road network weights value [12]. Yang Zhiyong, Zhao cold, Zhang Xingchen and other scholars studied the traveller routing choice model under a priori information, and the model studied the routing selection under prior information, not the traveller gets real-time information [13]. Zhang He, Yangzhao Sheng, Wang Wei and other scholars used the method of increasing load to ensure the optimal routing, and also used the predicted road traffic flow decomposed into K parts proportionally, each part calculates travel time by BPR function, and determined the shortest routing by Dijkstra algorithm [14]. G.Pang, K.Takahashi and other scholars on basis of fuzzy neural network method studied adaptive routing in dynamic routing selection system, through fuzzy neural network selection training driver's routing selection, establishing driver routing selection function, the function is adaptive [15]. GAO Feng used simulation method to establish the modeling of driver's routing choice behavior, this model is based on decision theory and Bias's theory, the simulation results show that the decision theory and Bias theory are favourable for drivers' dynamic routing selection behavior [16].

In view of existing shortcomings of traditional Dijkstra algorithm, this paper proposed an pre-treatment optimazation strategy based on date constraints and weight optimization strategy based on routing impacts factors to improve traditional algorithm's speed and accuracy in traffic routing planning.

2 The deficiency of Dijkstra algorithm

Dijkstra algorithm is a shortest routing algorithm based on greedy strategy; its main idea is to construct a tree path in accordance with the growth of tree routing length point by point method, thus obtained the shortest routing from the root node of the tree (namely, the starting point) to all other nodes.

Assuming set S storages shortest path that has been calculated; in the initial state, set S only have one source point v_0. After seek one shortest routing $(v_0, v_1, ..., v_k)$, and then add v_k to the set S until all vertices have joined in S.

Introduce an auxiliary vector d, each component d_i, represents the shortest routing length that can be found from the source point v_0 to other vertices v_i. Its initial state:

If exists arc from v_0 to v_i, then d_i is weight value on arc, or d_i is ∞, set the first shortest routing is (v_0, v_k), and k satisfies the equation (1).

$$d_k = \min_i \{d_i \mid v_i \in V - v_0\} \tag{1}$$

Then the next shortest routing (the terminal point is v_j), or (v_0, v_j), (v_0, v_k, v_j). In general, assuming S is the set of terminal point which has calculated shortest routing, then next shortest routing's intermediate node's length is:

$$d_k = \min_i \{d_i \mid v_i \in V - S\} \tag{2}$$

After obtained one shortest routing, added the terminal point v_k into the set S, then all the other vertices $v_i \in V - S$ will modify into d_i. See equation (3).

$$d_k = \min\{d_i, d_k + c(v_k, v_i)\} \tag{3}$$

In equation, $c(v_k, v_i)$ is weight value in arc (v_k, v_i). The above algorithm will produce the shortest path that from source point to other vertexes. For the single vehicle routing planning, only need to compute a shortest routing from source point to destination, so it can simply modify the algorithm, namely, when the shortest routing is determination of target routing, and algorithm will terminate. According to the above principle, a weighted directed graph $G = (V, E)$, which contains a vertex is the vertex set, contains arcs arc set, is from the arc, arc non negative weights, V is in the vertexes set which has n vertexes, E including m arc' set, (v, w) is the arc from v to v in E, $c(v, w)$ is the nonnegative weight of (v, w), assuming s is the vertex in V, t is vertex which can through s arrive the destination, then the shortest routing searching process which has the minimum arc weight value from s to t is as follows:

(1) Distributing $k(v)$, $d(v)$, $p(v)$ to each vertex v, in which $k(v)$ is a Boolean variable indicates whether or not the vertex v 's shortest routing has obtained; $d(v)$ is the shortest routing length's upper bound from s to v that is known, $p(v)$ is v 's rear vertex pointer, and initializing them respectively:

$$d(v) = \begin{cases} 0, v = s \\ \infty, v \neq s \end{cases} \tag{4}$$

$$k(v) = false \tag{5}$$

$$p(v) = \{\} \tag{6}$$

(2) Scanning $k(v) = false$'s vertices, from which selects a vertex v has a minimum routing length, then

$$k(v) = true \tag{7}$$

$$d(v) = \min\{d(v_i) \mid k(v_i) = false, v_i \in V\} \tag{8}$$

(3) Detecting each vertex v which $k(w) = false$ and adjacent to vertex v, if meets

$$d(w) > d(v) + c(v, w) \tag{9}$$

Then,

$$d(w) = d(v) + c(v, w), p(w) = v \tag{10}$$

(4) Repeating the scanning operations (2) and (3), until $k(t) = true$.

(5) From t to traverse the rear vertex pointer p source point s, the shortest solution will be obtained.

$$P_{st} = \{v_0 = s, v_1, ..., v_k = t\} \tag{11}$$

$v_i = p(v_{i+1})$, $i = 0, 1, ..., k-1$.

As can be seen from the algorithm processes, the algorithm has the following deficiencies:

(1) In classical Dijkstra algorithm, it needs to re-compare and re-modify all the nodes in S in searching for one point, which wasted a lot of time. For massive amount of data query, the waiting time makes it difficult to endure;

(2) The Dijkstra algorithm must be completely through every point.

(3) This method of adjacency matrix storage space, also wastes a lot of space in routing searching, because it must record all the nodes' information.

3 The improvement of Dijkstra algorithm

3.1 PRETREATMENT OPTIMIZATION BASED ON DATA CONSTRAINT

To reduce the number of nodes' expanding, it pre-treats all directed node pairs in the traffic routing network, and increases the optimal mark property, which records whether directed node pairs arrived at all blocks' optimal way. Directed node pairs' optimal marks are shown in Figure1.

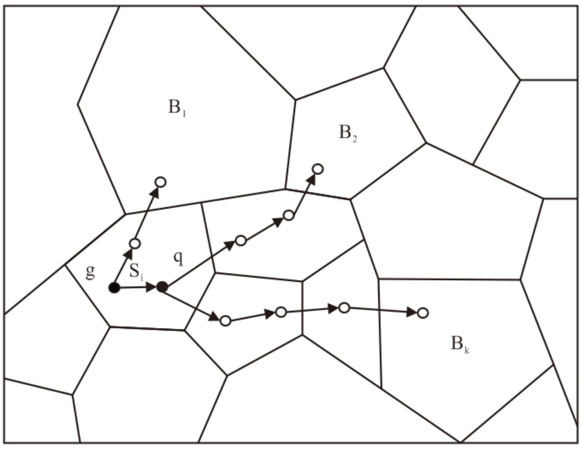

FIGURE 1 There are signs to the node on the optimal schematic

As shown in figure1, g , q are any two connected nodes in network, the positive number is s_i , the closed graph composed of black thick lines is the block in traffic routing network, light-colored thick line indicates the optimal routing from node g to each block. If the node q is on the optimal routing of which node g to each block, then the optimal node pair' optimal mark is true.

Calculate the optimal mark that from s_i to all blocks, its node to $\overrightarrow{gq}$ corresponding optimal node pairs' set is $F_{(i)}$. $f_{k,i}^{(f)}$ is the optimal mark which directed node through positive connection s_i arrives to objective block B_k. The size of each mark is 1, the node's optimal marks are arranged in number of blocks, $N(B)$ is the number in traffic routing network. The optimal mark of the node is shown in equation (12).

$$f_{k,i}^{(f)} = \begin{cases} 1 \\ 0 \end{cases} \tag{12}$$

In equation (12), 1 indicates in optimal routing of which from node s_i to block B_k, or not.

3.1.1 The boundary nodes' restrictions

Assuming t_j is the boundary node of block B_k, $N(B,k)$ is the number of boundary nodes, the optimal routing from node g to block B_k can be represented the optimal routing of which from this mode to all nodes in the blocks.

Block nodes are divided into boundary nodes and internal nodes, the node g to block internal node's optimal routing must passing through boundary nodes, so the optimal routing which node g to block can be represented as node g to all boundary nodes' optimal routing. The optimal routing set $P_{g,k}$ of which from node g to boundary nodes is shown in Equation (13).

$$P_{g,k} = [P_{g,k,j}], (j = 1, 2, ..., N(B,k)) \tag{13}$$

In equation (13), $P_{g,k,j}$ is number j 's optimal routing from node g to block, $P_{g,k,j}$ is the node pairs serial number set , see equation (14).

$$P_{g,k,j} = \{s_i\} \tag{14}$$

Then the calculation of optimal mark $f_{k,i}^{(f)}$ in equation (12) can be represented as equation (15).

$$f_{k,i}^{(f)} = \begin{cases} 1, s_i \in P_{g,k} \\ 0, else \end{cases} \tag{15}$$

Assuming that there has m modes in traffic routing network, each node has k positive connections, then the calculation of directed nodes in network to all the blocks' optimal marks needs $m \times k$ times Dijkstra expansion, the time required is too long the expansion method, so that this expansion method' time is too long, here we take block as the basic processing unit, adopts reverse Dijistra expansion to require nodes' optimal mark, it only needs expand n times to obtain all nodes' optimal mark.

Node t_j on the basis of Dijkstra method to process reverse expansion, in expansion procession, it records each expanding node's expansion attribute q , as shown in equation (16).

$$q = \{q_b, q_a, q_d\} \tag{16}$$

In equation (16), q_b is the block serial number of expansion node q , q_a is node q 's ID number; q_d is the minimum cost from node to node t_j .

3.1.2 Optimal node's mark optimization

When all nodes in traffic network are expanded, then the node t_j 's reverse expansion will stop. The node g has multiple rear connected nodes; the equation (17) determines whether the node pair $\overrightarrow{gq}$ is inevitable routing to reach to the target node t_j . If satisfies equation (17), it means that directed connection s_i is on the optimal routing to node t_j, then the objective node's optimal mark $f_{k,i}^{(f)}$ which from directed s_i to boundary node t_j. See equation (17).

$$f_{k,i}^{(f)} = \begin{cases} 1, g_d - q_d = w(g,q) \\ 0, g_d - q_d > w(g,q) \end{cases} \tag{17}$$

$w(g,q)$ is the cost from node g to node q. The optimal mark $f_{k,i}$ from directed node $\overrightarrow{gq}$ to block B_k is shown in equation (18).

$$f_{k,i}^{(f)} = f_{k,i,1}^{(f)} \cup ... \cup f_{k,i,j}^{(f)} \cup ... \cup f_{k,i,N(B,k)}^{(f)} \tag{18}$$

As mentioned above, through reverse breadth expansion of boundary node in target block B_k can obtain the optimal mark of each section to this block; similarly, through positive expansion can obtain the reverse optimal mark of which each directed node to this block. The optimal routing mark $f_{k,i,j}^{(r)}$ of which from directed node pairs $\overrightarrow{gq}$ to target node t_j, and the reverse optimal road signs $f_{k,i}^{(r)}$ of which from node to target block B_k, see equation (19) and (20).

$$f_{k,i,j}^{(r)} = \begin{cases} 1, g_d - q_d = w(g,q) \\ 0, g_d - q_d > w(g,q) \end{cases} \tag{19}$$

$$f_{k,i}^{(r)} = f_{k,i,1}^{(r)} \cup ... \cup f_{k,i,j}^{(r)} \cup ... \cup f_{k,i,N(B,k)}^{(r)} \tag{20}$$

These methods can be calculate the optimal mark $f_{k,i}$ and $f_{k,i}^{(r)}$ which from directed node to any target blocks, if the optimal mark takes directed node as unite for storage, then in the process of routing searching, when we expand to one node it will read files, which causing I / O time-consuming increased, and the efficiency rate of data's reading is not high.

In order to solve the above problem, here we will storage the optimal mark as "target block", it only exist two blocks in the process of planning: starting block and terminal block, then we only need to read one optimal routing mark, thereby reducing reading time.

All blocks' optimal marks set F in traffic routing network is shown as equation (21).

$$F = [f_k], (k = 1, 2, ..., N(B)) \tag{21}$$

In equation (21), f_k is k's optimal mark set, including forward optimal mark set $f_k^{(f)}$ and reverse optimal mark set $f_k^{(r)}$, see equation (22).

$$f_k = \{f_k^{(f)}, f_k^{(r)}\} \tag{22}$$

Forward optimal mark set $f_k^{(f)}$ and reverse optimal mark set $f_k^{(r)}$ are respectively shown in equation (23) and (24).

$$f_k^{(f)} = [f_{k,i}^{(f)}], i = 1, 2, ..., N(s) \tag{23}$$

$$f_k^{(r)} = [f_{k,i}^{(r)}], i = 1, 2, ..., N(s) \tag{24}$$

$N(s)$ is the number of all directed node pairs in traffic routing network. The storage way of the optimal routing set F is for random. In random reading model which all directed nodes – blocks, the size of block optimal mark f_k is fixed, it uses sequential storage, block optimal mark set f_k's internal nodes optimal marks $f_{k,i}^{(f)}$ and $f_{k,i}^{(r)}$ also use the same way for storage. which can achieve routing mark's random positioning and reading, for the calculation of the optimal routing marks, make use of a single row way for storage, namely, when node optimal mark $f_{k,i}^{(f)}$ or $f_{k,i}^{(r)}$ is "true", then the calculation of this node pair's optimal mark.

3.2 WEIGHT OPTIMIZATION BASED ON ROUTING IMPACTS FACTORS

In a given map, some routing have been described and according it to find the optimal routings that already exist, the traditional Dijkstra algorithm is usually according to equation (3) to modify the distance which the source node v passes through k but not exists in node j of s.

Firstly, assuming that routing effected factor is $\Gamma()$, the algorithm's specific process is to add routing effected factors of d_j and $d_k + c(v_k, v_i)$ to equation (3). See equation (3).

$$d_k = \min\{d_i + \Gamma(v, j), d_k + c(v_k, v_i) + \Gamma(v, k, j)\} \tag{25}$$

In the course of the iteration, the number of routing edges varies with the changes of routing, so according to equation (25) to select the routing, when records d_k's value, we do not add the routing affected factors. Because in an iterative process, the routing is constantly changing, so it is necessary to record the edges number. How to solve the problem of recording the edges number when the current node passes through the shortest routing, we can set up an $N[]$ array in improved algorithm, whose role is to store each iteration's number of edges, and in each iteration to solve weights cumulative sum which can add routing effected factors.

See equation (26).

$$d_k = \min\{d_i + \Gamma(N[j]), d_k + c(v_k, v_i) + \Gamma(N[k]+1)\} \tag{26}$$

When the routing changes, we modify $N[j]$ value:

$$N[j] = N[k]+1 \tag{27}$$

Thus the current optimal routing passing edges' number, for which from source node to other nodes can through $N[]$ to record.

Then, assuming that w has a node, and considering from m to n's optimal routing problem, in the original routing, the optimal routing from m to n must go through node k, under the condition that it passes through k, the distance from m to n is much higher than the distance that passing through w. to the elapsed down to the distance.The traditional Dijkstra algorithm usually according to the equation (10) to modify the distance from source node m to node n.

According to the same principle, first assuming that routing effected factor is $\Gamma()$, and algorithm's specific implementation steps is adding $d_k + c(v_k, v_i)$ and $d_k + c(v_w, v_n)$'s routing effected factors to equation (10). See equation (28).

$$d_k = \min\{d_k + c(v_k, v_n) + \Gamma(m,k,n), d_w + c(v_w, v_n) + \Gamma(m,w,n)\} \tag{28}$$

The edging number which from source node to the other nodes can use $N[]$ to correctly record, shown as equation (26) ~(27).

4 Algorithm performance simulation

In order to verify the effectiveness of the improved algorithm which proposed in this paper, we will have the simulation experiments.Firstly, using different nodes' traffic network tests algorithm's computing speed, the simulation results are as follows:

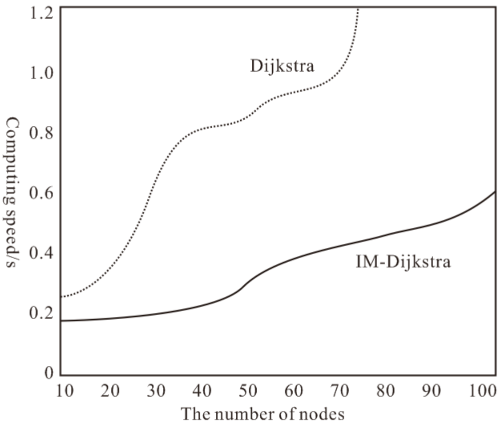

FIGURE 2 Nodes in different paths algorithms computing speed

As we can seen from the figure, the improved Dijkstra algorithm which proposed in this paper, its operation speed is much faster than the traditional Dijkstra algorithm. Then test shortest routing's accuracy on different nodes' traffic routing, the accuracy statistics are as follows:

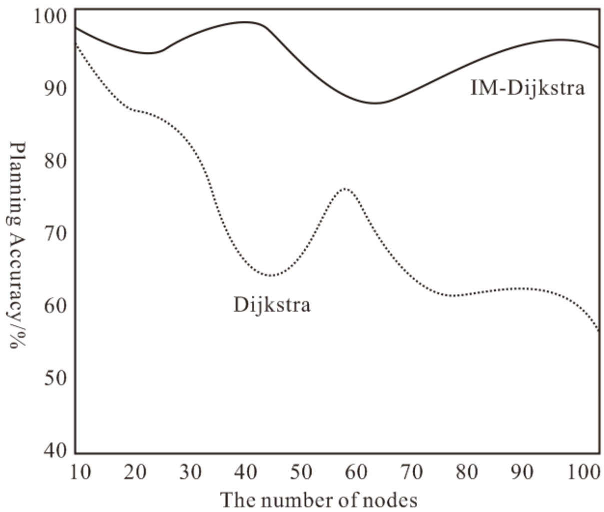

FIGURE 3 Different path planning algorithm node accuracy

As can be seen from the above simulation results, compared to traditional Dijkstra algorithm, Dijkstra algorithm based on data constraints and routing effected factors which proposed in this paper has a higher speed and accuracy in traffic routing planning.

5 Conclusions

However, road congestion is more and more serious along with the growth of vehicles, the optimal routing planned by the existing routing planning system cannot meet people's needs, people hope it will be a system that can consider a variety of constraint information's routing planning system, which can consider of current traffic conditions, rather than static routing alone. This paper pre-treated and optimized the traditional Dijkstra algorithm, and obtained an improved traffic routing planning model, the simulation results show that this model owns a faster operation speed and a higher accuracy, and also owns a good robustness.

References

[1] Yu Zhaoyuan (2014) Multi-type constraint network optimal routing analysis algorithm based on geometric algebra, *Geographic and Geo-information science*, **30**(2), 10-15.

[2] Cai Qiang (2013) Routing weight determination method based on the grey neural network, *Journal of Shanghai university of science and technology*, **35**(6), 552-556.

[3] Zhang Kai (2013) Bus Travel Algorithm Based on Vector Matching in Sparsely Populated Areas. *Transportation Systems Engineering and Information*. **13**(6), 127-133.

[4] Guo Dahong (2013) Self path planning and management method for unidirectional guided-path based automated guided vehicle system. *Manufacturing Automation*, **35**(11), 98-100.

[5] Zhang Shuaiyi, (2013) Optimal path planning method under the restriction of cognitive. *Science of surveying and mapping*, **38**(3), 157-159.
[6] LI Lurong (2013) A dynamic path planning model based on the optimal ant colony algorithm, .*Journal of Guangxi University（Natural Science Edition）*, (2), 359-367.
[7] HU Jihua (2013) Hierarchical Path Planning Method Based on Taxi Driver Experiences. *Transportation Systems Engineering and Information*, (1), 185-192.
[8] Lan Zhikun, (2012) Multiple AGV system of dynamic path planning algorithm, *Highway traffic science and technology*, **29**(10), 121-125.
[9] Li Xun (2012) Study of vehicle dynamic safety path planning method under nternet of things. *Journal of Northwestern Polytechnical University*, **30**(5), 729-733.
[10] Liu Zhiqi (2012) The study on urban traffic optimal path planning simulation. *Computer Simulation*, **28**(7), 349-352.
[11] Liang Dong (2012) Intelligent vehicle path planning and control algorithms under semi-structured road. *Transportation systems engineering and information technology*, **11**(2), 44-51.
[12] Sun Yuyu (2012) Heuristic function based on the traffic confining rule and road level factors, *Journal of Dalian Maritime University*, **36**(4), 55-58.
[13] Liu Zhanglei (2012) A Dynamic Routing Planning Strategy Based on Road Network Changes. *Transportation Systems Engineering and Information*, **10**(3), 147-152.
[14] Tang Liang (2012) Path planning method based on improved ant colony algorithm, *Journal of University of Science and Technology of China*, (9), 980-983.
[15] Wu Jing (2012) A Network Model for Multimodal Routing Planning. *Journal of National University of Defense Technology*, **31**(5), 81-85.
[16] Li Longji (2012) Path Planning in Complex Transportation Network Based on Nonlinear Circuit. *Information and Contro*, **38**(4), 425-430.

Authors

Jun Yuan, 23. 6. 1986, China

Jun Yuan received the Master's Degree in transportation planning and management from Wuhan University of Technology, China in 2011. Currently, he is a researcher at Wuhan University of Technology, China. His research interests include Transportation Planning and The new logistics.

Peilin Zhang, 2. 1955, China

Peilin Zhang received the PHD in Management Science and Engineering from Huazhong University of Science and Technology, China in 2000. Currently, he is a Doctoral tutor at Wuhan University of Technology, China. His research interests include Transportation Planning and Logistics Management.

Liqun Ding, 3. 1983, China

Liqun Ding received the Master's Degree in transportation planning and management from Wuhan University of Technology , China in 2010. Currently, he is a researcher at Wuhan University of Technology, China. His research interests include Container Transport and The new logistics.