Available online at www.sciencedirect.com
ScienceDirect
ELSEVIER
Procedia - Social and Behavioral Sciences 96 (2013) 1069 – 1078
Procedia
Social and Behavioral Sciences

13th COTA International Conference of Transportation Professionals (CICTP 2013)

A Totally Astar-based Multi-path Algorithm for the Recognition of Reasonable Route Sets in Vehicle Navigation Systems

Wei YIN[a*], Xiaoguang YANG[a]

a .School of Transportation Engineerin, Tongji University, No. 4800 Cao'an Road, Jiading District, Shanghai,601804, China

Abstract

Compared with a Dijkstra-based or partially Astar-based one, a totally Astar-based algorithm is proposed in the paper for vehicle navigation systems. It has a better performance such as computing speed and veracity in a large-scale road network than a Dijkstra-based one because the computational complexity of Astar algorithm has little connection with the overall scale of a road network. To recognize all the reasonable routes between a specific OD pair, this algorithm takes all the geometrically reasonable routes into account and considers several constraints that meet the drivers' preferences like circuitous route, the number of turns and traffic control strategy (for example, no left turn). Two numerical examples demonstrate the operation and efficiency of the algorithm.
Keywords: Multi-path Algorithm, A-star, Route navigation systems

Keywords: Multi-path Algorithm, A-star, Route navigation systems

1. Introduction

The reasonable route set is defined as a set of all paths that drivers may choose between a specific OD pair and it is more meaningful than a single shortest path in dynamic vehicle navigation systems. Given uncertainty of link travel time and interference of on-trip events, a fixed shortest path does not exist in a road network. And delivering multiple paths instead of single ones is a feasible way to handle the drivers' different preferences and to avoid the overload of specific paths.

Traditionally, multi-path algorithms can be divided into two types: the k-shortest path algorithms proposed by Eppstein (1998) and Martins et al (1999), and disjoint path algorithms proposed by Dinic (1970) and Torrieri (1992). Chen et al (2007) indicate the drawbacks of k-shortest path algorithm that it becomes inefficient for finding alternative paths when the number of paths between an OD pair is large. And they design a partially

doi:10.1016/j.sbspro.2013.08.123

Astar-based algorithm to solve this problem. Bell (2009) designs a multi-path Astar algorithm called Hyperstar for vehicle navigation with risk aversion.

The pathfinding algorithm used in a navigation system has a high request in computing speed because the algorithm is carried out frequently to find changeable real-time optimal paths due to the random on-trip events. LuFeng et al (1999) set an oval restricted area on the network prior to the multi-path search by the k-Dijkstra algorithm. The area is efficient because the number of nodes traversed by algorithm is decreased. Wagner and Willhalm (2006) review the methods for speeding up pathfinding algorithms and summarize two classical speed-up techniques, which are bidirectional search and goal-directed search (Astar algorithm).

In summary, the main factor influencing the efficiency of algorithm is the number of nodes that have to be traversed. LuFeng et al (1999) use a prior area to reduce it but the drawback of the method is that some feasible paths may be lost and the border is hard to define because some circuitous paths are allowed. And directional search mentioned by Wagner and Willhalm (2006) use directional potentials to decrease it automatically. For example, Astar algorithm introduces node potentials to modify the priority of the nodes in Dijkstra's open list, thereby raising the efficiency of search Bell (2009).

Deletion Algorithm (J.A.Azevedo et al, 1990) finds a shortest path, deletes one link on it and search for another reasonable path, thereby organizing a multiple paths set. The drawback of it is that some similar paths and loops may exist in the results. And Barra (1993) finds a new reasonable path by increasing the impedances of links on the current reasonable path.

Drivers' preference, which is another factor, must be considered. Szczerba et al (2000) discuss the setting of constraints for various mission scenarios in a pathfinding algorithm. In the paper, the constraints on drivers' preference are listed as follows, maximum path duration, minimum path reliability, maximum number of turns and maximum ratio of minor road to the entire path length.

In this paper, a new multi-path algorithm is designed which integrates deletion algorithm and Astar algorithm to weaken the negative affection of large-scale network, considers the drivers' preferences and traffic control strategies and allows some reasonable detours.

The remainder of this paper is organized as follows. In Section 2, we give the key procedure of the algorithm and describe the constraints. In Section 3, two numerical examples are designed. The example 1 expresses the operation of the algorithm in a small network. And the example2 shows the efficiency of it in a relatively large grid network. Finally some conclusions are presented in Section 4.

2. Algorithm framework

2.1. Algorithm layout

To express the algorithm, a road network consisting of a set of links A and a set of nodes I is considered. And the following sets and variables are defined.

A	Set of links
A_{ab}	The link from node a to node b
I	Set of nodes
$I_i^{\#}$	Set of adjacent nodes of node i

f_i^{ab} The flow comes into the node i from node a and go out to node b
F_f Set of forbidden f_i^{ab}
A_f Set of forbidden links, $A_f \in A$
t_{ij} The number of times that link from node i to node j is chosen.
s_i Linear distance between node i and end node
d_i Linear distance between node i and start node
k_i The number of circuitous links in one path of which the end point is node i
k_m The maximum number of circuitous links in one path
u_{ab} Travel time of A_{ab}
wm_i Least travel time from start node to node i
w_i Travel time from start node to node i
con parameter of path type
H Set of nodes attached to possible circuitous links
G Set of nodes attached to normal links

Variables have been initialized prior to the algorithm:

$H \leftarrow \emptyset; G \leftarrow \text{start point},$

$con \leftarrow 0; t_{ij} \leftarrow 0, i,j \in I;\ k_i \leftarrow 0, i \in I; wm_i \leftarrow \text{INFINITY};$

As is shown in Fig.1, the algorithm begins from the start node and then chooses some adjacent node as next node iteratively (section 2.2). If the node is the end node, a new path formed by all chosen nodes adds into paths set. And then the algorithm comes back to start node to search a next path. If a path doesn't satisfy corresponding constraints (section 2.3), it breaks and starts a new path search from the start node. The algorithm terminates when the iterative limit is satisfied.

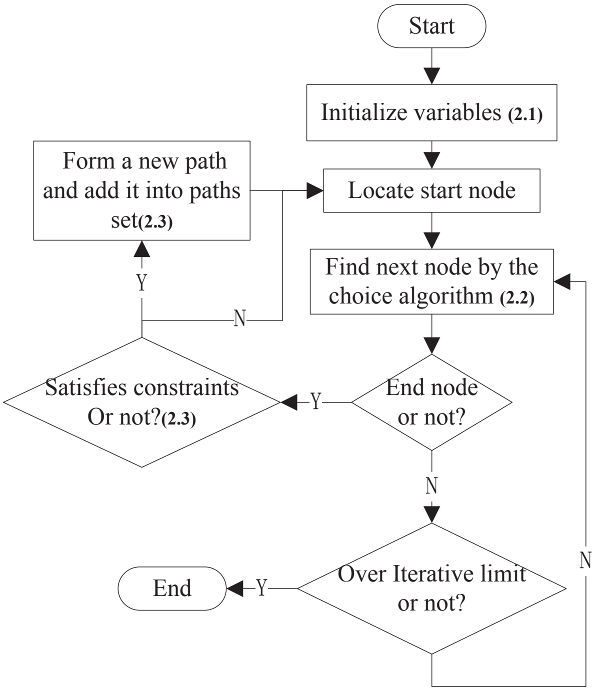


Fig.1 Layout of the algorithm

2.2. Iterative choice of next node

The iterative choice of next node is a key component of the algorithm because a new path which contains those nodes must be reasonable and different from the others in the paths set.

As is shown in Table 1, the choice algorithm is divided into four layers whose priority rank from high to low. If the upper layer keeps neutral, the lower one makes a decision.

Table 1 four layers of decision-making

Priority	Objective
LevelA	Generate a restricted area
Level B	Eliminate forbidden flows
Level C	Find a different path
Level D	Search the next link with high contribution to reach the target

Define I_c as current node, I_n as next chosen node and I_p as previous node. And the following algorithm shows the procedure of next node choice.

0. Level A:

#1 $i \leftarrow I_c$; $j \leftarrow I_p$; $L \leftarrow I_i^{\#}$; $z \in L$; $ap \leftarrow A_{ji}$;

#2 If $s_z - s_i \geq 0$ and $d_z - d_i \geq 0$ then $k_z \leftarrow k_i + 1$;

#3 If $s_z - s_i < 0$ *and* $d_z - d_i \geq 0$ then $k_z \leftarrow k_i$

#4 If $s_z - s_i < 0$ *and* $d_z - d_i < 0$ then $k_z \leftarrow k_i - 1$;
#5 If $(s_z - s_i \geq 0$ and $d_z - d_i < 0)$ *or* $k_z > k_m$ then $L \leftarrow L - \{z\}$;

1. Level B:
#6 if $L = \emptyset$ then $\{z\} \leftarrow \emptyset$, go to Step 4
#7 $k \leftarrow I_n, I_n \in L$;
#8 $an \leftarrow A_{ik}; f_c \leftarrow f_i^{ap,an}$;
#9 if $an \in A_f$ or $f_c \in F_f$ then $L \leftarrow L - \{I_n\}$ go to Step 1 else go to Step 2

2. Level C:
#10 find $z \in L$ with minimum t_{iz} ; $L \leftarrow \{z\}$;
#11 if the num of nodes in $\{z\} = 0$ then $t_{iz} \leftarrow t_{iz} + 1$; $I_n \leftarrow$ start point;
#12 if the num of nodes in $\{z\} = 1$ then go to Step 4
#13 if the num of nodes in {z}>1 then go to Step 3

3. Level D:
#14 Find $z \in L$ with maximum $(s_i - s_z)/u_{ab}$
#15 If $s_z - s_i < 0$ *and* $d_z - d_i \geq 0$ then $G \leftarrow G + \{z\}$; go to Step 4; //a normal link;
#16 If $(s_z - s_i) * (d_z - d_i) > 0$ then $H \leftarrow H + \{z\}$; $con \leftarrow 1$; go to Step 4

4. Get next node
#17 $I_n \leftarrow \{z\}$;
#18 $t_{iz} \leftarrow t_{iz} + 1$;
#19 $w_z \leftarrow w_i + u_{iz}$;
#20 $wm_z \leftarrow$minimum w_z;
#21 If $con \leftarrow 1$ and $I_n \in G$ and $w_z > wm_z$; then $I_n \leftarrow$ start point;con=0;

Conventional Astar Algorithm is the combination of directional potentials and Dijkstra Algorithm. In this paper, directional potentials are integrated with Deletion Algorithm and two potentials s_i d_i are introduced into the algorithm to recognize the type of links. Drives are supposed to be logical and believe that the next chosen road should make them more close to the target and some detours are allowed if they can reach the target faster. The potentials of which the inequality relation divides three kinds of conditions describe such actions as is shown in Fig.2. A normal link is preferred because such link has a direct contribution for drivers to reach the destination. A possible circuitous link is chosen if no normal link exists. And an invalid link is ignored in any condition.

L is the set of possible next nodes of I_c. And the element z (each possible next node) in L is eliminated by the choice of four levels until I_n is found. The Level A is used to form a restricted area, which is different from an artificial geometrical area mentioned in LuFeng et al (1999). The formation of the area is dynamic and non-geometrical. As is shown in Fig.3, the border of a reasonable area (dark gray) and circuitous area (light gray) is formed gradually with the iteration of nodes because the invalid links and possible circuitous link over the limit of k_m are eliminated in each node. And the border of the area is calculated by directional potentials and k_m so the area is not just a simple shape. The vital importance of area formation raises the computing speed by controlling the number of nodes especially in a large-scale network.

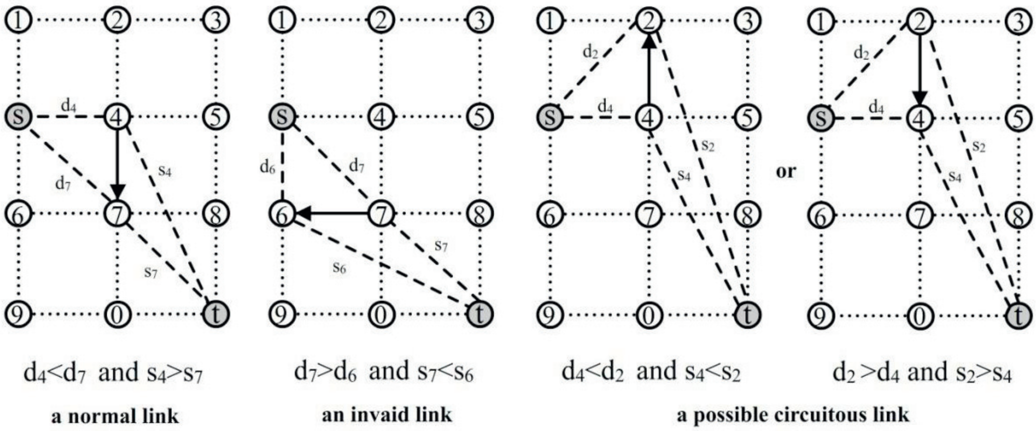


Fig.2 direction-oriented definitions of links

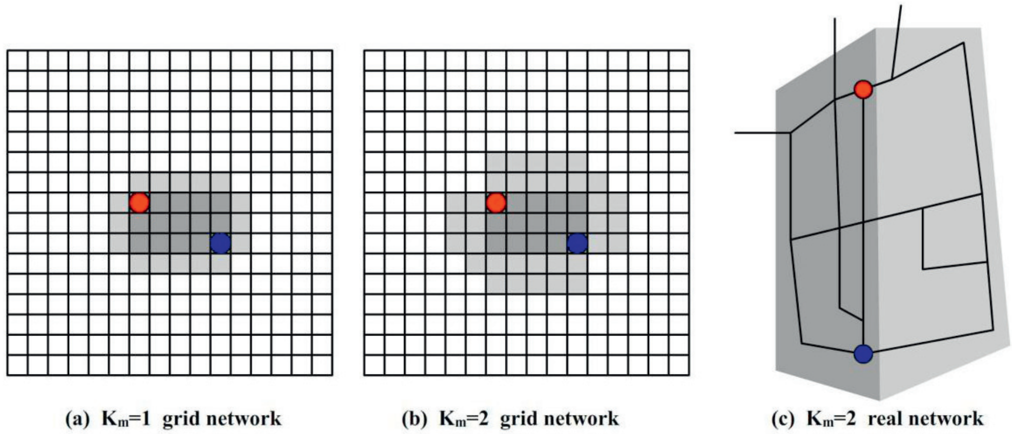


Fig.3 restricted area generated by directional potentials

Level B eliminates all the forbidden flows F_f and links A_f caused by the accident, road maintenance or traffic control policies like “No turn left”.

In Level C, T_{ab} is a key parameter which records the search frequency of A_{ab} and only links with minimum t_{ab} are allowed in L (#10). Such method is similar to the Delete Algorithm (J.A.Azevedo et al, 1990). The difference is that all the links of the current path are labeled (#18) instead of labeling one of them to find a next one. As one kind of breadth first search, it can traverse nodes rapidly and decrease the similarity of paths. The label could also be replaced by the impedance of roads, which is discussed in section 4, If more than one node is existed in L after Level C, I_n is decided by Level D.

Drivers prefer high grade roads, which is describe by the design speed. This variable is adapted and a new index $(s_i - s_z)/u_{ab}$ is proposed in Level D. The meaning of the length of a road is replaced by the decrease of linear distance contributed by the road. The algorithm assumes that drivers choose the road with a higher index (#14).

A normal path (con=0) is defined as a path consisting of normal links. And a detour (con=1) must contain any possible circuitous links. When a circuitous link with negative index is chosen, the condition is added that the circuitous path has a shorter travel time than that of any normal one (#15, 16, 19~21).

2.3. Path constraints

The drawback of this algorithm is that the iteration of a next node may create some invalid, inefficient or overlapping paths. So a series of path constraints is proposed in the paper which includes maximum path duration, minimum path reliability, maximum number of turns and maximum ratio of minor road to the entire path length.

The maximum number of paths in the reasonable paths set is determined first. And then a constraints system is established, which depends on specific environment (Szczerba et al ,2000). If a path fails to satisfy any constraints, it is eliminated and a new path search starts. If it is more excellent after comparing with other paths in the set, it will replace the worst one.

3. Numerical examples

3.1. Example1

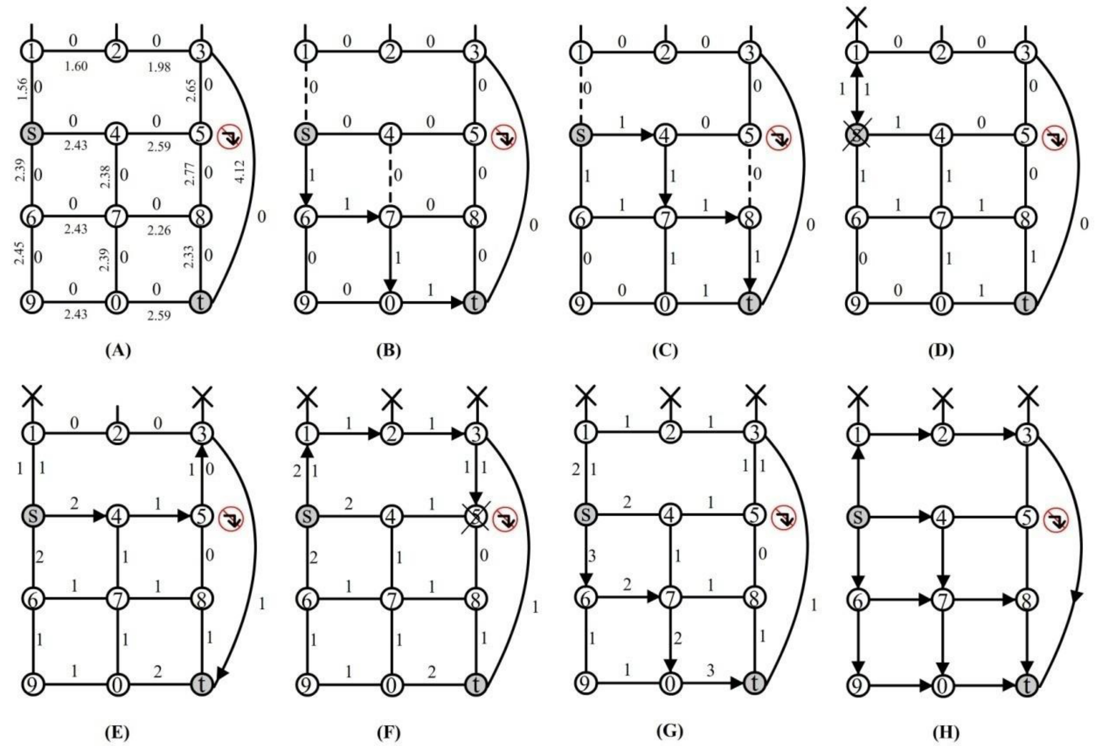


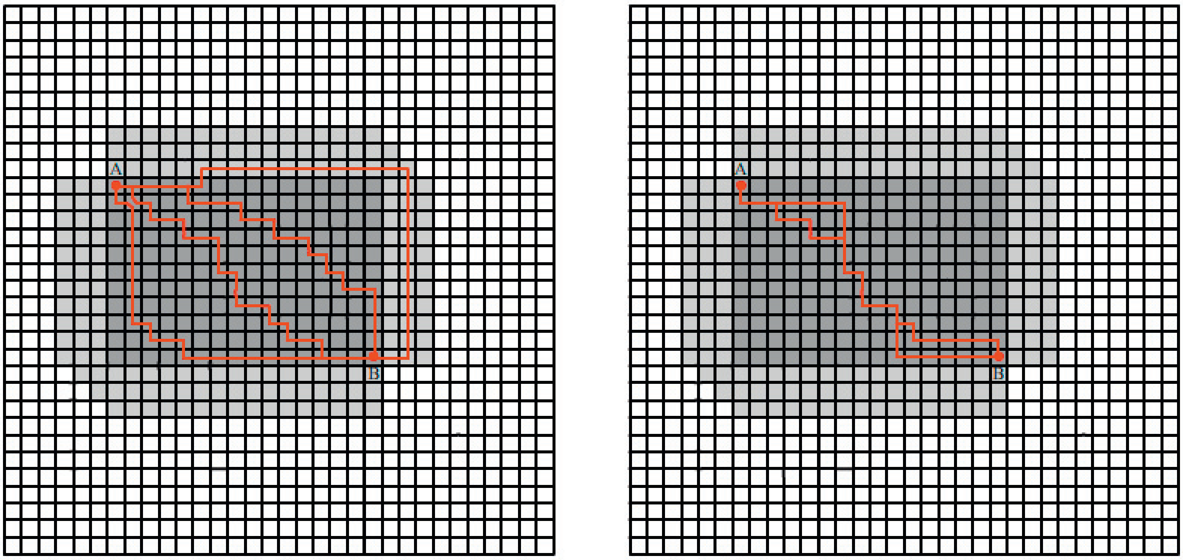
Fig.4 the numerical example of the algorithm

To simplify the presentation, we only show a key network consisting of 12 nodes and 17 links , omit the others by the radial lines connected with Node 1, 2,3 and set k_m to 1. The t_{ij} of all links are initialized to 0. And the travel time of each link is labeled (see Fig.4A). The start and end node are “s” “t” and one forbidden flow exists in Node 5. Maximum path duration is considered and is set to the value of the fourth shortest path.

According to the algorithm, the choice of first path depends on Level D. And the second one is totally decided by Level C due to the labels of first one (see Fig.4B,C). And then a possible circuitous link A_{s1} is chosen and node 1 is set to the current node so A_{1s}, A_{1x}, A_{s2} are three candidates of next link. A_{1x}, which connects with the other part of the network, is eliminated in Level A because $k_x > k_m$ and becomes a border of the restricted area. A_{1s} is chosen due to a higher index in Level D but the entire path breaks because the travel time of the circuitous path ($s \rightarrow 1 \rightarrow s$) is longer than the path ($s \rightarrow s$) (see Fig.4D).Such situation also occurs in Fig.4F. In Level B $f_5^{4,8}$ is a forbidden flow so Node 3 is chosen as the next node when the algorithm calculates the Node 5. And the path ($s \rightarrow 4 \rightarrow 5 \rightarrow 3 \rightarrow t$) becomes a reasonable circuitous path (see Fig.4E). An odd situation that one path may be found repeatedly occurs in Fig.4G , which is discussed in next section. All the reasonable paths are presented in Fig.4H. And the path ($s \rightarrow 4 \rightarrow 5 \rightarrow 3 \rightarrow t$) is eliminated because it fails to satisfy path constrains.

3.2. *Example2*

To test the efficiency of the algorithm, a randomly generated grid network of 1024 nodes is discussed (see Fig.5). One cell symbolizes one intersection. And the real coordinates of it and the travel time of corresponding links are random in a finite interval to simulate a real road network. The start and end node are respectively “A” “B” and set k_m to 3. Four optimal paths are required in this test.



Multi-path A-star based algorithm K-Dijkstra based algorithm

Fig.5 the calculated results of a large grid network

Table 2 Results Analysis of two algorithms

Items	Mean value of travel time(min)	Number of Iterations	Variance
Multi-path A-star	56	10000	8.94
K-Dijkstra	49	10.4 million	3.25

As is shown in Fig.5 and Table 2, the results of the multi-path Astar based algorithm is prominent in the independence among chosen paths and the number of iterations. However, the optimality of the chosen paths is lower than that of K-dijkstra one.

4. Discussion

In future, more people will install navigation devices to handle complicated and changeable traffic environment. It leads to the risk that people may encounter a new kind of congestion due to the same path choice. The essence of this problem is that the pathfinding algorithm is simple and the calculation is mutually independent. To solve the above-mention problem, future vehicle navigation system may centralize navigation demands from drivers, calculate together and deliver different navigation information. The algorithm is proposed against this background because it satisfies two basic requirements of future navigation systems which are high computing speed and multiple reasonable paths.

The computing speed of the proposed algorithm rises significantly because the strategy of the algorithm is different from that of conventional multi-path algorithm. Original K-Dijkstra Algorithm whose complexity is O(n3) searches all paths and then compares them to find a reasonable path set. It traverses all nodes each time and cannot get results until termination so it is inefficient in both spatial and time dimension. And the improvement of the algorithm cannot essentially raise efficiency. However, the proposed algorithm use goal-direction search which decrease the number of calculated nodes. And most suboptimum or optimum result is found in the initial stage of iteration. The algorithm increases the efficiency in both sides.

Updating search frequency of links and updating impedance of links are two methods to get different paths. The first method as a breadth-first search can traverse nodes rapidly and increase the reliability of paths but many inefficient and repetitive paths are found. And the second one as a depth-first search can get more reasonable paths but some key links may lose and the reliability of paths decreases. The second one will be discussed in further works.

The drawbacks of the algorithm is the same as most heuristic method that it lacks a steady process of optimization and a specific convergence condition and the results are not strictly optimal. However, finding a set of suboptimum paths with relatively little iteration are more practical than searching optimum paths set with huge calculation in modern vehicle navigation system.

Acknowledgements

This research is supported in part by the Ministry of Science and Technology of the People's Republic of China(Grant No. 2011AA110404), the National Natural Science Foundation of China (Grant No. 60974093), Science and Technology Commission of Shanghai Municipality (Grant No. 10530705100). The writers wish to thank Dr Yue Liu From University of Wisconsin – Milwaukee and Dr Wanjing Ma from Tongji University for their helpful suggestions and useful comments on an earlier version of the paper.

References

Chen, Y., M. G. H. Bell, et al. (2007). "Reliable pretrip multipath planning and dynamic adaptation for a centralized road navigation system." Intelligent Transportation Systems, IEEE Transactions on 8(1): 14-20.

Eppstein, D. (1998). "Finding the k shortest paths." SIAM J. Comput. 28(2): 652-673.

Martins, E. Q. V., M. M. B. Pascoal, et al. (1999). "Deviation algorithms for ranking shortest paths." International Journal of Foundations of Computer Science 10(3): 247-262.

E. A. Dinic (1970). "Algorithm for solution of a problem of maximum flow in a network with power estimation," Sov. Math.Dokl. 11. 248–264.

D. Torrieri, (1992). "Algorithms for finding an optimal set of short disjoint paths in a communication network," IEEE Trans. Commun., 40(11): 1698–1702.

Bell, M. G. H. (2009). "Hyperstar: A multi-path Astar algorithm for risk averse vehicle navigation." Transportation Research Part B: Methodological 43(1): 97-107.

LuFeng ,LuDongmei ,CuiWeihong(1999)."Time Shortest Path Algorithm for Restricted Searching Area in Transportation Networks" Journal of Image and Graphics,4(10): 850-853.

Szczerba, R. J., P. Galkowski, et al. (2000). "Robust algorithm for real-time route planning." Aerospace and Electronic Systems, IEEE Transactions on 36(3): 869-878.

J.A.Azevedo, J.J.E.R.S.Madeira, E.Q.V.Martins and F.M.A.Pires (1990). "A shortest paths ranking algorithm" Proceedings of the Annual Conference AIRO'90, Models and Methods for Decision Support, Operational Research Society of Italy,1001-1011.

De La Barra, T., B. Perez, et al. (1993). "Multi-dimensional path search and assignment." PTRC-PUBLICATIONS-P: 307-307.