# 論文 / 著書情報
# Article / Book Information

| | |
|---|---|
| Title | Finding multiple dissimilar reliable routes in linear time complexity under travel time uncertainties |
| Authors | Jiangshan MA, Daisuke FUKUDA |
| Citation | TRB 94th Annual Meeting Compendium of Papers, , Number 15-4376, |
| Pub. date | 2015, 1 |
| Note | The paper was peer-reviewed by TRB and presented at TRB 94th Annual Meeting, Washington, DC, 1, 2015. |

# Finding Multiple Dissimilar Reliable Routes in Linear Time Complexity under Travel Time Uncertainties

**Jiangshan Ma**
*Department of Civil Engineering, Tokyo Institute of Technology*
M1-11, 2-12-1, O-okayama, Meguro-ku, Tokyo 152-8552, Japan
Phone: +81-3-5734-2577, Fax: +81-3-5734-3578
ma.js@plan.cv.titech.ac.jp

**Daisuke Fukuda**
*Department of Civil Engineering, Tokyo Institute of Technology*
M1-11, 2-12-1, O-okayama, Meguro-ku, Tokyo 152-8552, Japan
Phone: +81-3-5734-2577, Fax: +81-3-5734-3578
fukuda@plan.cv.titech.ac.jp

Words: 5789
Figures: 8 (2000 words)
Tables: 0 (0 words)
Total words: 7789

**ABSTRACT**
There are considerable literatures on route travel time reliability and most of which are based on the stochastic characteristics of traffic networks. This paper studies reliable routing under travel time uncertainty with limited information available. Since *a-prior* shortest path (SP) may incur higher delays in an uncertain network, the route travel time reliability is considered to seek for reliable routes. Due to complexity and data availability, many sophisticated reliable routing algorithms are not suitable for large scale applications. In this paper, an algorithm finding several dissimilar routes ranked by their reliability indices based on a risk-averse hyperpath is proposed. For being searched on the pre-calculated hyperpath, which is a directed acyclic graph (DAG), the path-finding can be completed with a linear complexity.

# INTRODUCTION

Shortest path (SP) problems have been well studied in the past decades and widely applied to vehicle navigation. Algorithms used in commercial navigation systems are summarized in Hofmann *(1)* and Flinsenberg *(2)*. A* based algorithms *(3)* are dominant in finding the optimal path in most navigation systems. In transportation applications, a comprehensive review of heuristic shortest path algorithms can be found in Fu *(4)*.

Due to the nature of uncertainty in travel time, travelers may suffer unexpected long delays on the path recommended by SP algorithms. In recent years, some online map providers (e.g. Google Maps) or navigation systems (e.g. Toyota G-BOOK) started to deploy multiple routes and/or reactive en-route routing.

Fu and Rilett *(5)*, Miller-Hooks and Mahmassani *(6)* tried to find expected shortest path assuming link travel time distributions are known. Mean-variance based travel time reliability researches are found in Small *(7)* and further early/late arrivals are penalized in scheduling models *(8, 9, 10)*. Chen et al. *(11)* introduced a K-paths algorithm which takes path set reliability into consideration. In the context of vehicle routing, Kaparias et al. *(12)* formulated reliability indices for links and routes based on mean-variance method with the assumption that link travel times are lognormally distributed.

There are some issues on dealing with travel time uncertainty by providing the "most reliable" route. Firstly, distributional assumptions of travel times may different from real ones. Secondly, the assumption that travel time distribution on each link is accurately available is unrealistic in practice. Additionally, the randomness captured by probabilistic models is merely a part of uncertainty and some uncertainties (e.g. being disrupted by earthquake or congested due to traffic incidents) cannot be captured. For these reasons, instead of imposing the distributional assumptions, some studies focus on providing a set of alternative links with limited travel time information. Bell *(13)*, among others, argued that an optimal strategy hyperpath model could be transplanted to route guidance by assuming that road links also provide a kind of service of which frequency can be considered as the reciprocal of potential maximum delay on link. In this way, only the expected travel time and a potential maximum delay for a link are required for route calculation, which makes it possible to be applied to large-scale networks. Although applying hyperpath to route guidance is very promising on uncertain networks, the assumption of “link service frequency” is still weak in behavioral explanations because the assumption of headway distribution in original optimal strategy model remains unclear. Since a hyperpath is a potentially optimal link set substantially, Fonzone et al. *(14)* defined Potentially Optimal (PO) link as a link being involved in a revealed shortest path in any network states and proposed to find a PO set first and then search a prior route by regret-based decision making procedures, which is similar to the idea of efficient paths proposed by Dial *(15)*.

Routing algorithms with sophisticated models dealing with travel time distributions as well as link correlations may work well. However, the lack of data has made these algorithms difficult to apply, especially for large-scale networks. In contrast, this paper focuses on path-finding on uncertain networks with limited travel time information and aims to find *a-prior* dissimilar reliable routes efficiently by constraining the search within a pre-calculated risk-averse hyperpath. In this way, heavy computation and light computation can be separated for combining centralized and decentralized routing. The hyperpath plays its role as a potentially optimal link set which enables en-route switching or adaptive routing in the background of navigation terminals. The drivers are able to find several dissimilar reliable paths out of the hyperpath and maintaining the adaptability of the selected path. In case of foreseeable large delay, the driver may awake the background hyperpath and considering route swithing.

The rest of this paper is organized as follows. Section 2 introduces the risk-averse hyperpath model based on the work of Bell *(13)* and Ma et al. *(16)*. Section 3 proposes a reliability index for ranking the reliable routes and a cosine similarity index based method to find dissimilar paths. In section 4, a GIS platform with Client/Server (C/S) architecture based on open sources is introduced. The proposed algorithm is implemented on the platform and several numerical tests are conducted. Section 5 concludes this paper.

## RISK-AVERSE HYPERPATH UNDER TRAVEL TIME UNCERTAINTY

A real optimal route on uncertain networks cannot be determined beforehand since it relies on revealed travel times. A feasible method is to find a potentially optimal link set and conduct a link-to-link strategic link choice with revealed travel times. Similar ideas have occurred in Miller-Hooks and Mahmassani *(6)*, Yang and Miller-Hooks *(17)* and Gao and Chabini *(18)*, which deal with stochastic networks with known travel time distributions. Differently, following Bell *(13)*, we assume only lower and upper bounds of link travel time are available while the distributions are unknown, which excludes the possibility of using conventional stochastic methods. The lower and upper bounds are time-dependent, which results in different ranges over time slices. Since a static routing algorithm can be easily extended to time-dependent (with speed profiles), the static case is the focus of this paper. Based on Bell *(13)*, the risk-averse hyperpath problem in road network context is further illustrated in Ma et al. *(16)*.

The concept of hyperpath originates from Spiess and Florian *(19)* and Nguyen and Pallottino *(20)* in frequency-based transit assignment fields. A hyperpath is described by selecting attractive links based on optimal strategy that minimizing the total travel time including node waiting time. It can be considered as a result of sequential link choices. Let $H$ denote the link set of a hyperpath and $A_i^+$ denote the outgoing links from node $i$. Nökel and Wekeck *(21)* summarized different hyperpath models and concluded that finding optimal hyperpath is equivalent to finding the attractive link set $H_i^+$ ($H_i^+ = H \cap A_i^+$) for each node and then the probability $P_a$ of selecting a link $a$ out of $H_i^+$. For any link $a \in H$, let $W_a$ denote the waiting time for the first-coming vehicle on link $a$ and $s_k$ denote the remaining total travel time of a strategy using link $k$, the expected total travel time $T_a$ of link $a$ is

$$T_a = W_a + \sum_{k \in H} P_k s_k \quad [1]$$

In Spiess and Florian *(19)*, with the assumption of exponential distributions for vehicle headway and uniform passenger arrival, the combined node waiting time and link choice probability are noted as

$$W_i = \frac{1}{\sum_{k \in H_i^+} f_k} \quad [2]$$

$$P_a = \frac{f_a}{\sum_{k \in H_i^+} f_k} \quad [3]$$

where $f_a$ is the transit line service frequency and $W_i$ is the expected waiting time at node $i$ for any attractive link including link $a$. Then we have $W_i = W_a, \forall\, a \in H_i^+$ and $T_a$ then becomes

$$T_a = W_i + \sum_{k \in H_i^+} P_k s_k = \frac{1 + \sum_{k \in H_i^+} f_k s_k}{\sum_{k \in H_i^+} f_k} \quad [4]$$

Consequently, the behaviour of considering more attractive lines results in a larger combined frequency and hence less expected waiting time. As a matter of fact, the hyperpath is a result of optimal strategy towards travel time uncertainty and this also applicable to road networks. The hyperpath is a general idea of a potentially optimal path/link set out of links in a network that is involved with uncertainty. It has been widely used in fields including transit assignment, relational data bases *(22)* and traffic assignment in capacitated networks *(23)*.

Now we consider the routing context and see how the hyperpath can be adapted. In the context of routing, each intersection in can be regarded as a decision node where a driver has to make a link choice. Since all possible turns are not attractive to drivers, an attractive link set $H_i^+$ can be distinguished from all available links. Let $P_{a|i}$, $P_a$ for short, denote the conditional link choice probability in $H_i^+$ at a node $i$ in contrast to marginal link choice probability $p_a$ with respect to all links in the hyperpath $H$. Assuming that risk-averse travellers fear the maximum delays $d_a$ which may occur on links and always follow a strategy which takes links probabilistically, we have

$$\frac{P_k}{P_a} = \frac{d_a}{d_k}, \forall a, k \in H_i^+ \text{ and } k \in H_i^+ \quad [5]$$

The simple behaviour assumption can be generalized with behaviour models such as discrete choice models. However, the generalization is beyond this paper and thus we stick to this assumption in accordance with Bell *(13)*. Together with Eq. [5], since $\sum_{k \in H_i^+} P_k = 1$, the following equation which is mathematically equivalent to Eq. [3] can be derived if we replace $1/d_a$ with $f_a$.

$$P_a = \frac{1}{\sum_{k \in H_i^+} \frac{d_k}{d_a}} = \frac{\frac{1}{d_a}}{\sum_{k \in H_i^+} \frac{1}{d_k}} \quad [6]$$

Eq. [6] is exactly in line with Bell *(13)* referring to "nominated link frequency" that is though difficult to explain in road networks. The "expected waiting time" $W_i$ in a road network context can be interpreted as "average delay exposure" to the potentially occurrence of delays on downstream links. We use $e_i$ instead of $W_i$ to distinguish the context difference. With the assumption that more alternative links will help mitigate the degree of the exposure feelings to downstream maximum delays, the interpretation of risk-aversion becomes clear by rewriting Eq. [2] as

$$e_i = \frac{\sum_{k \in H_i^+} P_k d_k}{K_i} = \frac{\sum_{k \in H_i^+} \frac{1}{d_k} \bullet d_k}{K_i \bullet \sum_{k \in H_i^+} \frac{1}{d_k}} = \frac{1}{\sum_{k \in H_i^+} \frac{1}{d_k}} \quad [7]$$

where $P_k d_k$ denotes the exposure to maximum delay on link $k$ and $K_i$ is the number of attractive links at decision node $i$. If there is only one attractive link $a$, a driver will be fully exposed to the maximum delay $d_a$. By defining the delay exposure at nodes, $T_a$ then becomes the travel time anticipated by pessimists who fear maximum delay of $d_a$ on each link $a$. Let $p_i$ denote the node choice probability so that we have $p_a = P_a \cdot p_i,\ a \in H_i^+$. Then the risk-averse hyperpath problem under travel time uncertainty is formulated as

$$\min \sum_{a \in A} c_a p_a + \sum_{i \in I} p_i e_i$$
$$s.t. \begin{cases} \sum_{a \in A_i^-} P_a = 1, i \neq r \\ \sum_{a \in A_i^+} P_a = 1, i \neq s \\ P_a \propto \frac{1}{d_a}, \forall a \in A \\ p_i = 1, i = r \text{ or } i = s \\ p_a \geq 0, P_a \geq 0, p_i \geq 0 \end{cases} \quad [8]$$

Let $\omega_i = \sum_{a\in H_i^+} p_a \cdot e_i = p_i e_i$, which denotes the expectation of delay exposure at decision node $i$, Eq. [8] becomes the linear program described in Bell *(13)*, which is an equivalency to that in Spiess and Florian *(19)*, albeit being applied to different context. Now that the hyperpath problem for transit networks and road networks turns out to be of the same formulations with different interpretations, the risk-averse hyperpath problem can also be solved with existing algorithms such as Spiess & Florian algorithm *(19)*, Hyperpath-Dijkstra algorithm *(24)*, Hyperstar algorithm *(13)* or $\mathrm{SF^{di}}$ algorithm *(16)*. The computation time of these algorithms are also compared in Ma et al. *(16)*.

## HYPERPATH-BASED RELIABLE ROUTING

The hyperpath provides *a-prior* link-to-link routing strategy so that a complete route consisting of multiple link choices is attainable after reaching the destination. The hyperpath routing does not actually present any prior route recommendations but makes en-route turn recommendations. In case of real-time information is unavailable, the link-to-link choice will be fully made by drivers themselves since drives are believed to be able to know information by other means (e.g. Variable Message Signs, F.M. radio). Furthermore, if real-time travel time information is available in the system-side, advanced real-time routing model based on hyperpath can be developed. However, this is not the case in this paper. In most cases, network-wide real-time data are unavailable, which is the status quo in most cities even though engineers have been struggling in collecting more and more data. Consequently, before putting a routing method which can take advantages of both historical and real-time data into use, we propose to use a complementary method of hyperpath routing. The routing method should take advantages of limited available historical information and be efficient as well. Due to the simplicity and applicability, we focus on hyperpath-based routing and propose the method of providing several dissimilar reliable routes as pre-trip route alternatives. The proposed solution is deemed as a compromise among behaviour reality (verses conventional shortest path), computational requirement (verses sophisticated models) and data availability (verses data-driven routing models) and can be an alternative to currently used navigation systems. The general procedure of hyperpath-based routing is illustrated in FIGURE 1.

INSERT FIGURE 1 HERE

Firstly we find the hyperpath and then find multiple dissimilar paths with cosine similarity index (CSI) on the generated hyperpath. Finally these paths are ranked by the order of the proposed reliability index (RI). Details are explained in the following.

### Reliability index under travel time uncertainty

On uncertain networks with only limited information of link travel time bounds (namely the minimum and the maximum travel time), it is impossible to apply statistics based indices such as in Lam and Small *(26)*, van Lint and van Zuylen *(27)* and Kaparias et al. *(12)*. Mean, variance and/or quantiles are necessary to these studies. However, none of these studies can deal with the case when only the maximum and minimum travel times are known. Utilizing the limited information, the reliability index of a $k^{th}$ route $\delta_k$ consisting of $n_k$ links with travel time uncertainty is simply defined as Eq. [9].

$$RI_{\delta_k(1,2,\ ,n_k)} = \frac{\sum_{i=1}^{n_k} t_i}{\sum_{i=1}^{n_k} X_i \cdot \sum_{i=1}^{n_k} \frac{d_i}{t_i}} \qquad [9]$$

where

$$X_i = \begin{cases} 0, & d_i < \beta_i \cdot t_i \\ 1, & \text{others} \end{cases} \qquad [10]$$

$t_i$ and $d_i$ are optimistically anticipated (the minimum) link travel time and the maximum delay of link $i$ respectively. The sum of $X_i$ (penalty to very large delays) represents the total fear of large delays and $\beta_i$ in Eq. [10] is defined as a tolerance factor which differentiates the acceptable and unacceptable delays. The sum of optimistically anticipated travel time as the denominator in Eq. [9] refers to the relative attitude towards delay. It is reasonable that the same level of delay is more bearable for longer trips than for short trips. The sum of relative delay to travel time reflects a general way in which people make decisions. The absolute reliability becomes positive infinity if there is no delay and reduces to 0 when delays approach positive infinity.

**Finding dissimilar routes**

In single-objective routing problems, it may be enough to select a single "best" path. On the other hand, routing on uncertain networks is a multi-objective routing problem with the criteria such as route travel time, reliability and dissimilarity. It is usually necessary to generate a set of paths and evaluate them under the relevant criteria. Conventional methods evaluating path dissimilarities are based on link length overlap *(28, 29, 30)*. Unlike these studies, we take advantages of cosine similarity index in vector space model that is commonly applied to text similarity comparison *(31)*. This method has also been used for path similarity comparison in Ma and Fukuda *(32)* and Fonzone et al. *(14)*. On preparing multiple paths, following K-shortest path algorithms in De la Barra *(33)* and Lim *(29)*, the heuristic K-shortest path (KSP) algorithm which has been implemented by incremental path weights is used: all links on the shortest path will be increased after finding a shortest path in an iteration, which continues until K paths have been found.

The following two conditions should be satisfied when deciding to add a new route during the K-shortest path process with respect to path similarity:

(a) The route has never been added in previous iterations.

(b) The route is not "similar" to any other route that has been added.

Then, let $\boldsymbol{V}_a$ denote the vector of route $a$ defined by

$$\boldsymbol{V}_a = [x_1 l_1,\ x_2 l_2, \quad ,\ x_i l_i] \qquad [11]$$

where $x_i$ equals to 0 if a link $i$ is not included in path $a$ and 1 otherwise, and $l_i$ is the length of link $i$. The Cosine Similarity Index (CSI) between two routes $a$ and $b$ is defined as

$$CSI_{ab} = \frac{\boldsymbol{V}_a \cdot \boldsymbol{V}_b}{||\boldsymbol{V}_a|| \cdot ||\boldsymbol{V}_b||} \qquad [12]$$

For example, consider a simple sub-network shown in FIGURE 2, the number in brackets are link identifications and others are link lengths. There are two possible routes $r_1 (1 \to 2 \to 4)$ and $r_2$ $(1 \to 3 \to 4)$ from node $A$ to $B$, the corresponding vectors are $\boldsymbol{V}_{r_1} = (10, 12, 0, 18)$ and $\boldsymbol{V}_{r_2} = (10, 0, 15, 18)$ and consequently $CSI_{r_1 r_2} \approx 0.698$.

INSERT FIGURE 2 HERE

If a newly generated route is not similar to any accepted route, accepting the route will help to decrease average similarity. Let $\overline{CSI_R}$ and $\overline{CSI_{R'}}$ denote average CSI of a route set before and after adding the newly generated path, the path search implements a heuristic criteria as follows

$$\overline{CSI_R} \,/\, \overline{CSI_{R'}} - 1 > \qquad [13]$$

where a parameter $\epsilon$ $(\epsilon > 0)$ is used to adjust acceptable extent of similarity. Larger $\epsilon$ implies more rigorous requirement on dissimilarity and results in smaller number of routes with larger dissimilarity among each other. Note that the $\overline{CSI_R}$ should have been defined based on comprehensive comparisons between any two routes within the route set as

$$\overline{CSI_R}^* = \frac{\sum_{r_i \in R}\sum_{r_j \in R} CSI_{r_i r_j}}{N_R^2} \quad [14]$$

where $N_R$ is the number of routes added to the reliable route set. Noting that only the changing items are concerned in Eq. [14], it can be simplified as

$$\overline{CSI_R} = \frac{\sum_{r_i \in R} CSI_{r_i r'}}{N_R} \quad [15]$$

where $r'$ is the newly added route.

**Risk-averse hyperpath based reliable routing algorithm**

The greatly important thing to apply a routing algorithm in practice is the computational efficiency. The hyperpath can be created within a time complexity of at most $O(n\log n + m\log\delta)$ *(24)*. With the optimistic node potentials and node directed search proposed in Ma et al. *(16)*, the actual computation time can be substantially reduced. Based on the above-mentioned contents, the algorithm for generating multiple reliable routes is illustrated as follows.

*Algorithm for Hyperpath Based Reliable Routing (HBRR)*

1: generate hyperpath $A'$ on network $G(I, A)$ by Procedure HP (hyperpath, see *13, 16, 25*);
2: create a sub-network topology $G'(I', A')$ for $A'$;
3: conduct Procedure KRSP (K Reliable Shortest Paths) shown below;
4: path sorting by RI.

*Procedure KRSP:*

$K$ – maximum number of iterations
$N$ – maximum number of routes
$\lambda$ – penalty factor, $t_a = t_a \cdot (1 + \lambda)$
1: **Initialization:** set $K, N, \lambda, R \leftarrow \emptyset, CSI_{\mathrm{R}} \leftarrow 0$;
2: **while** $i < K$ and $N_R < N$ **do**
3: find the route $r_i$ with the least undelayed travel time
4: **if** $i \notin R$ **then**
5: calculate $V_{r_i}$ with respect to $G'(I', A')$
6: **if** $R = \emptyset$ **then**
7: $R \leftarrow R \cup \{r_i\}$
8: $CSI_{\mathrm{R}} \leftarrow 1$
9: **else**
10: $CSI_{\mathrm{sum}} \leftarrow 0$
11: **for** each route $r_j$ in $R$
12: $CSI_{sum} \leftarrow CSI_{sum} + CSI_{r_i, r_j}$
13: **end for**
14: $\overline{CSI}_{R'} \leftarrow \frac{CSI_{sum}}{N_R}$
15: **if** $\frac{\overline{CSI}_R}{\overline{CSI}_{R'}} - 1 > \epsilon$ **then**
16: $\overline{CSI}_R \leftarrow (CSI_{sum} + 1)/(N_R + 1)$
17: $R \leftarrow R \cup \{r_i\}$
18: **end if**
19: **end if**

```
20:     end if
21:     penalize travel time on each link of r_i by λ
22: end while
```

Lines 6 – 8 add the first-found SP to R, since there is only one route in R, the cosine similarity of a route compared with itself is 1 (line 8). Line 14 calculates current average similarity by comparing the newly generated route to each existing routes in R. Lines 15 – 18 update the average similarity and add a new route to R if Eq. [13] is satisfied. Otherwise, the newly generated route is ignored because it is judged similar to some existing routes. Once again we would like to emphasize that the word "similar" is subjectively defined by a dissimilarity parameter $\epsilon$. Since hyperpath is a potentially optimal link set leading to the destination whose size tend to increase with the level of delay on networks, when a network is slightly congested the routes found from the sub-network constructed by hyperpath tend to share many links in common so that we should define a more tolerant $\epsilon$ to find enough routes. On the contrary, if a network is quite uncertain, the hyperpath links may get more dispersed and larger $\epsilon$ should be adopted in order to find routes with more dissimilarity.

The hyperpath is always a directed acyclic graph*(34)*. As the multiple reliable path algorithm is actually based on limited iterations of shortest path algorithm, the time complexity can be reduced to linear for navigation terminals (in-vehicle navigation systems, smart phones etc.) on condition that the hyperpath is pre-stored.

## NUMERICAL TEST ON A GIS PLATFORM

### GeoRouting.Net

To implement the proposed algorithm on GIS environment, a light GIS platform is developed. With the help of several open source tools, we built a light platform named GeoRouting.Net under Microsoft .Net Framework. The platform is C/S based architecture which mainly relies on PostgreSQL, PostGIS and PgRouting installation on the server side and SharpMap project on the client side development.

PostgreSQL server is adopted as database for the reason that it has a good GIS support with extensions of PostGIS and PgRouting. Similar to the commercial ArcSDE of ESRI Company, PostGIS spatially enables the PostgreSQL server to serve as a backend GIS database saving and operating spatial data. Besides supporting Open Geospatial Consortium (OGC) spatial data types, about 800 functions (dependent on the specific version) are provided in PostGIS which makes it capable of various spatial operations such as spatial buffering and linear referencing. As a complement of PostGIS, PgRouting focuses on network analysis issues such as topology building which is used in GeoRouting.Net. Besides, some fundamental routing algorithms (e.g. Dijkstra's algorithm, A* algorithm, TSP algorithm etc.) are also provided. On the client side, we take advantage of SharpMap, which is a young open source library under GNU LGPL V2.1 license. It provides with map rendering, operation and publishing for both desktop and web development under .Net Framework. It supports various data format and provides drivers for many databases (e.g. PostGIS, SpatiaLite, SQL Server, ESRI Geodatabase). Many other open source libraries have been involved in SharpMap such as Proj4.Net and NetTopolgogySuite. SharpMap is powerful and light as well, although the core library files add up to only about 1 Mega. A light GIS platform named GeoRouting.Net is developed especially for routing research mainly based on above-mentioned tools. Some other open source libraries are also used for special purposes such as interacting with Google Earth.

GeoRouting.Net consists of two modules of route planning and map display shown in FIGURE 3 and FIGURE 4 respectively. The route planning module interacts with non-spatial data doing path calculations and updates path results in database. The tab pages irrelevant to this research are not shown in FIGURE 3 although the route planning module designed for general routing purposes.

INSERT FIGURE 3 HERE

The map display module mainly deals with spatial data visualizing spatial geometries. Common map operations such as zooming and panning are also provided. DB layers and Memory layers indicate data sources, Origin, destination and reliable routes are saved in memory in the form of well-known text (WKT) in the example shown in FIGURE 4. We see a hyperpath generated from New York State network provided by FHWA (*http://www.fhwa.dot.gov/planning/nhpn/2005/*) in which thicker and darker links are of higher chosen probability. As can be seen in "Memory layers", there are four reliable routes found by proposed algorithm with parameters in FIGURE 3.

INSERT FIGURE 4 HERE

**Numerical Tests**

Two parameters, namely $\epsilon$ and $\lambda$, are involved in the path search. A sensitivity analysis to these parameters will be investigated in a numerical test with a randomly selected OD pair on a 50*50 grid network (2500 nodes, 9800 links) with synthetic link travel time uniformly distributed within [0, 1] and different delay levels uniformly distributed within [0, 0.25γ] where γ denotes the level of delay. Default parameters are set to be $K = 1000$, $N = 20$, $\epsilon = 0.1$ and $\lambda = 0.1$. As for the RI calculation, we set $\beta_a = 0.5$, $\forall a \in A$ assuming delays larger than half of travel time may be unacceptable. FIGURE 5 shows hyperpaths on different delay levels (upper and lower nodes are destination and origin nodes respectively). Especially, FIGURE 5 (a) shows that when all links have no delay, the hyperpath reduces to the shortest path. This can be explained by the fact that Eq. [8] reduces to Eq. [16] which is similar to the shortest path problem when $\gamma = 0$. The only difference is that $p_a$ is limited to be dummy variable in shortest path problem but not in Eq. [16], which will cause multiple equal-possibility paths being provided by Eq. [16] when the costs of more than one routes are the same.

$$\min \sum_{a \in A} c_a p_a$$
$$s.t. \begin{cases} \sum_{a \in A_i^+} p_a - \sum_{a \in A_i^-} p_a = g_i \\ p_a \geq 0 \\ g_i = \begin{cases} 1, \text{if } i = r \\ -1, \text{if } i = s \\ 0, \text{otherwise} \end{cases} \end{cases} \qquad [16]$$

INSERT FIGURE 5 HERE

FIGURE 6 shows the reliable routes generated with respect to each scenario in FIGURE 5. Routes information are separated with underscores, the second item is route reliability calculated by Eq. [15] and the third item is route travel time without considering delay. In the case of route 1 in FIGURE 6(b), for example, the reliability index is calculated as 0.1 and the route travel time is about 10.43.

In FIGURE 6(a), since reliable routes are created on the sub-network constructed by the hyperpath which is equivalent to the shortest path, it is not surprising that only one reliable route with infinite reliability is found. More routes tend to be found as reliable paths in FIGURE 6(b) and (c) because the sub-networks from which they are created consist of more links. In FIGURE 6(b), route 4 is the one with least travel time estimation, however, route 1 is more preferred because of higher reliability index.

INSERT FIGURE 6 HERE

FIGURE 7 shows the effect of the size of $\epsilon$ on the number of reliable paths under the same delay level ($\gamma = 10$). As expected, it can be seen that larger $\epsilon$ results in less routes because of more rigorous requirement on dissimilarities. Another interesting feature in FIGURE 7 is that the number of reliable paths becomes less sensitive as $\epsilon$ increases. This may be explained by the fact that the remaining routes have stronger dissimilarities as $\epsilon$ increases and the same extent of increase in $\epsilon$ cannot easily change these routes. When $\epsilon > 6.2$, the number of reliable paths reduces to 1, which means that all other routes are considered similar to the first generated shortest path under this rigorous definition of dissimilarity.

INSERT FIGURE 7 HERE

Path search with incremental link weight is a heuristic method to find multiple paths in short time so that there may not any strict relation between $\lambda$ and the paths generated. Though, we also have tested the effect of $\lambda$ in tests with 37 different $\lambda$ in [0.1, 2] (interval = 0.1), [1, 10] (interval = 1) and [10, 100] (interval = 10). Test results are illustrated in FIGURE 8. Although no strict relation is found between $\lambda$ and the number of reliable paths, if the penalty is very large, the number of reliable paths tends to be stable according to tests with $\lambda \in [10, 100]$.

## CONCLUSIONS

This paper studied reliable routing problem with assumption of travel time information limited to the upper and lower bounds. An algorithm named HBRR which generates several routes concerning route reliability and dissimilarity is proposed. Unlike conventional routing algorithms, HBRR is based on a potentially optimal hyperpath with simple risk-averse behaviour assumptions and the reliable path search is actually conducted on a sub-network, which makes the searching process much faster in nature. Furthermore, the hyperpath is directed acyclic and can be calculated in advance, the multiple reliable paths can be found as fast as within linear time complexity if a pre-stored hyperpath is provided. Concerning the path similarity, the cosine similarity index taking routes as vectors weighted by link lengths simplifies the dissimilar path search and allows adjusting requirement on route dissimilarity by a setting parameters.

Though, parameters representing dissimilarity or reliability proposed in this paper are both comparative parameters targeted at path ranking and lack reality meanings. For example, it is still difficult to say how reliable a route with RI = 0.2 is. Although the RI is sufficient for route ranking, an absolute RI may be developed for purposes such as understanding the risk of taking the recommended path. Besides, this paper provides an alternative way of reliable route finding but the validation work has not yet done. More experiments for evaluating whether these *a-prior* routes are actually more reliable than conventional ones are required. Depending on the network scale which is concerned, more complex hyperpath models could be possible extended to.

The travel time in real traffic network is uncertain instead of simply stochastic and time-dependent, which causes algorithm development and validation more difficult. Considering the link-to-link choice as a sequential decision in a Markov Decision Process, some machine learning algorithms may help on this issue and it is still an open issue.

## REFERENCES

1. Hofmann Wellenhof B., Legat K. and Wieser M., *Navigation: principles of positioning and guidance*, 2003, Springer Verlag.
2. Flinsenberg, I., *Route planning algorithms for car navigation*, 2009, VDM Verlag.
3. Hart, P.E., Nilsson, N.J. and Raphael, B., A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions On Systems Science And Cybernetics*, Vol. 4, 1968, pp. 100 - 107.
4. Fu, L., Sun, D. and Rilett, L.R., Heuristic shortest path algorithms for transportation applications: state of the art, *Computer Operational Research*, Vol. 33, 2006, pp. 3324 - 3343.

5. Fu, L. and Rilett, L. R., Expected shortest paths in dynamic and stochastic traffic networks, *Transportation Research Part B: Methodological*, 1998, Vol. 32, pp. 499-516.
6. Miller-Hooks, E.D. and Mahmassani. H.S., Least expected time paths in stochastic, time-varying transportation networks, *Transportation Science*, Vol. 34, No. 2, 2000, pp. 198 - 215.
7. Small, K.A., Noland, R., Chu, X. and Lewes, D., Valuation of travel-time savings and predictability in congested conditions for highway user-cost estimation, *NCHRP Report 431*, Transportation Research Board, National Research Council, 1999.
8. Noland, R. B. and Small K.A., Travel-time uncertainty, departure time choice, and the cost of morning commutes. *Transportation Research Record*, Vol.1493, 1995, pp. 150 - 158.
9. Bates, J., Polak, J., Jones, P. and Cook, A. The valuation of reliability for personal travel, *Transportation Research Part E: Logistics and Transportation Review*, 2001, Vol. 37, pp. 191-229.
10. Fosgerau, M. and Karlström, A., The value of reliability. *Transportation Research Part B: Methodological*, Vol. 44, No. 1, 2010, pp. 38 - 49.
11. Chen, Y.Y, Bell, M.G.H. and Bogenberger, K., Reliable pre-trip multipath planning and dynamic adaptation for a centralized road navigation system. *IEEE Transactions on Intelligent Transportation Systems*, Vol. 8, No.1, 2007, pp. 14 - 20.
12. Kaparias, I., Bell, M.G.H. and Belzner, H., A new measure of travel time reliability for in-vehicle navigation systems. *Journal of Intelligent Transport Systems* 12(4), 2008, pp. 202 - 212.
13. Bell, M.G.H. (2009) Hyperstar: A multipath A-star algorithm for risk averse vehicle navigation. *Transportation Research Part B: Methodological*, Vol. 43, No. 1, 2009, pp. 97 - 107.
14. Fonzone, A., Schmöcker, J.D., Ma, J.S., and Fukuda, D., Link-based route choice considering risk aversion, disappointment and regret. *Transportation Research Record* Vol. 2322, No. 1, 2012, 119 - 128.
15. Dial, R., A probabilistic multipath traffic assignment model which obviates path enumeration. *Transportation Research*, Vol. 5, 1971, pp. 83 - 111.
16. Ma, J.S., Fukuda, D., Schmöcker, J.-D., Faster hyperpath generating algorithms for vehicle navigation, *Transportemetica A*: *Transport Science*, Vol. 9, 2013, pp. 925 - 948.
17. Yang, B. and Miller-Hooks, E., Adaptive routing considering delays due to signal operations, *Transportation Research Part B: Methodological*, Vol. 38, 2004, pp. 385 - 413.
18. Gao, S. and Chabini, I., Optimal routing policy problems in stochastic time-dependent networks, *Transportation Research Part B: Methodological*, Vol. 40, 2006, pp. 93 - 122.
19. Spiess, H. and Florian, M., Optimal strategies: A new assignment model for transit networks. *Transportation Research Part B*, Vol. 23, No. 2, 1989, pp. 83 - 102.
20. Nguyen, S. and Pallottino, S., Equilibrium traffic assignment for large-scale transit networks. *European Journal of Operational Research*, Vol. 37, 1988, pp. 176 - 186.
21. Nökel, K., and Wekeck, S., Boarding and alighting in frequency-based transit assignment. *Transportation Research Record*, No. 2111, 2009, pp. 60 - 67.
22. Gallo, G., Longo, G., Pallottino, S., Nguyen, S., Directed hypergraphs and applications, *Discrete Applied Mathematics*, Vol. 42, 1993, pp. 177 - 201.
23. Marcotte, P., Nguyen, S. and Schoeb, A.m A Strategic Flow Model of Traffic Assignment in Static Capacitated Networks. *Operations Research*, Vol.52, 2004, pp. 191 - 212.
24. Cominetti, R., Correa, J. and Correo, C., Common-lines and passenger assignment in congested transit networks. *Transportation Science*, Vol. 35, No.3, 2001, 250 - 267.
25. Bell, M. G., Trozzi, V., Hosseinloo, S. H., Gentile, G. and Fonzone, A. (2012) Time-dependent Hyperstar algorithm for robust vehicle navigation. *Transportation Research Part A: Policy and Practice*, Vol. 46, 2012, pp. 790 - 800.
26. Lam, T.C. and Small K.A., The value of reliability: measure from a value pricing experiment. *Transportation Research Part E*, Vol. 37, 2001, pp. 231 - 251.
27. van Lint, J.W.C. and van Zuylen, H.J., Monitoring and predicting freeway travel time reliability – using width and skew of day-to-day travel time distribution. *Transportation Research Record*, Vol. 1917, 2005, pp. 54 - 62.

28. Akgün, A., Erkut, E. and Batta, R., On finding dissimilar paths. *European Journal of Operational Research*, Vol. 121, 2000, pp. 232 - 246.
29. Lim, Y. and Kim, H. A shortest path algorithm for real road network based on path overlap, *Journal of the Eastern Asia Society for Transportation Studies*, 2005, Vol. 6, pp.1426 - 1438.
30. Zijpp, N.J. Van der and Catalano S. Fiorenzo, Path enumeration by finding the constrained K-shortest paths. *Transportation Research Part B: Methodological*, Vol. 39, No. 6, 2005, pp. 545 - 563.
31. Tan, P.-N., Steinbach, M. and Kumar, V., *Introduction to data mining*, Pearson Addison Wesley, Boston, 2006.
32. Ma, J.S., Fukuda D. Evaluating hyperpath based routing algorithm with taxi probe data, *Proceedings of the 16th HKSTS Conference*, 2011, pp. 123 - 130.
33. De la Barra, T., Perez T. B. and Anez, J., Multidimensional path search and assignment. *Proceedings of the 21st PTRC Summer Meeting*, 1993, pp. 307 - 319.
34. Nielsen, L. R., Andersen, K. A. and Pretolani, D. Finding the K shortest hyperpaths. *Computers & Operations Research*, Vol. 32, 2005, pp. 1477 - 1497.

**LIST OF FIGURES**

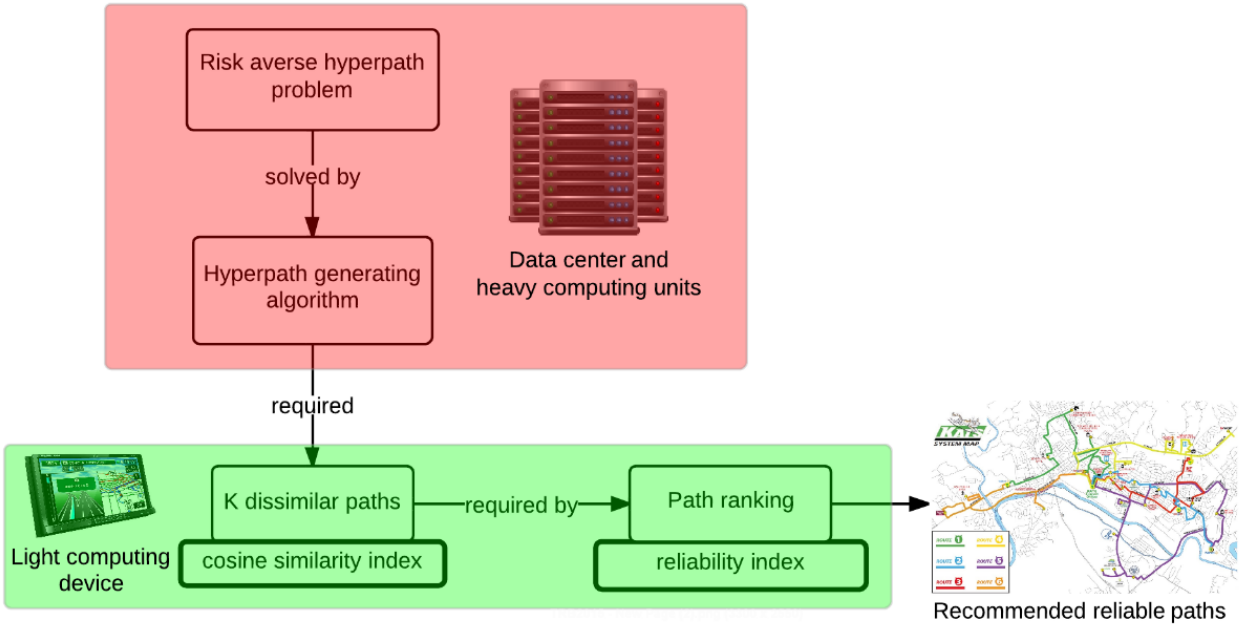



**FIGURE 1 Procedure of route recommendation**

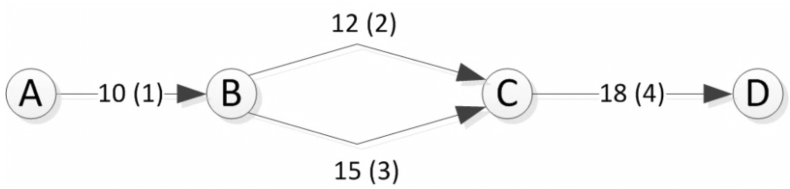



**FIGURE 2 A simple example sub-network constructed by hyperpath links**

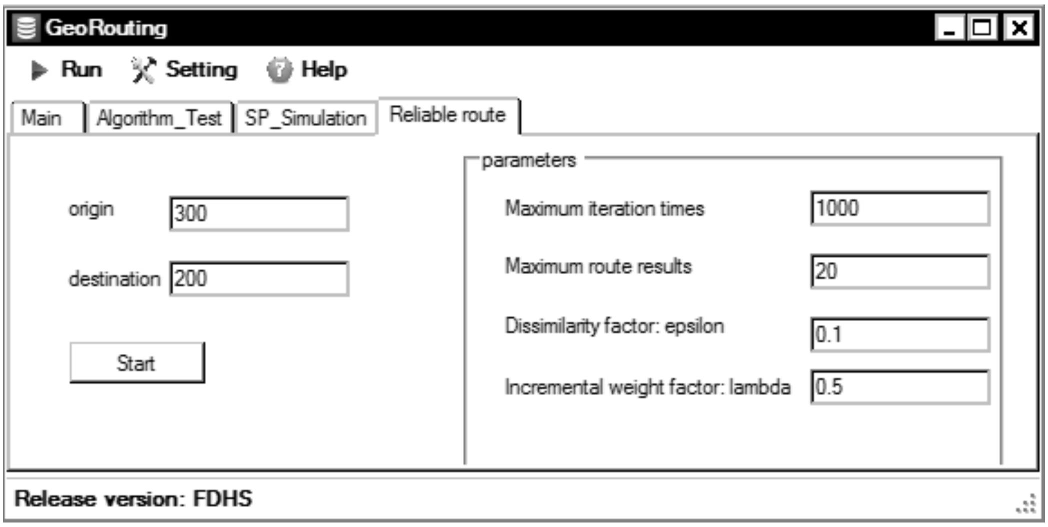



**FIGURE 3 Route planning module in GeoRouting.Net**

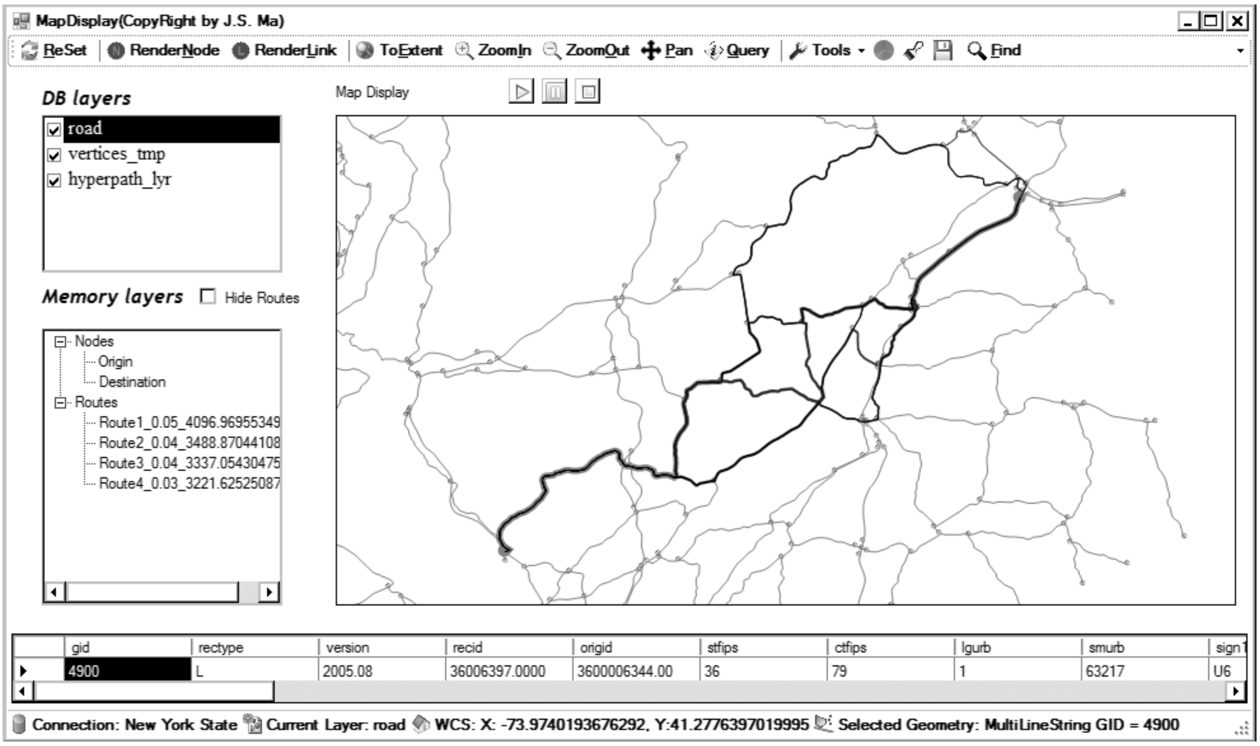
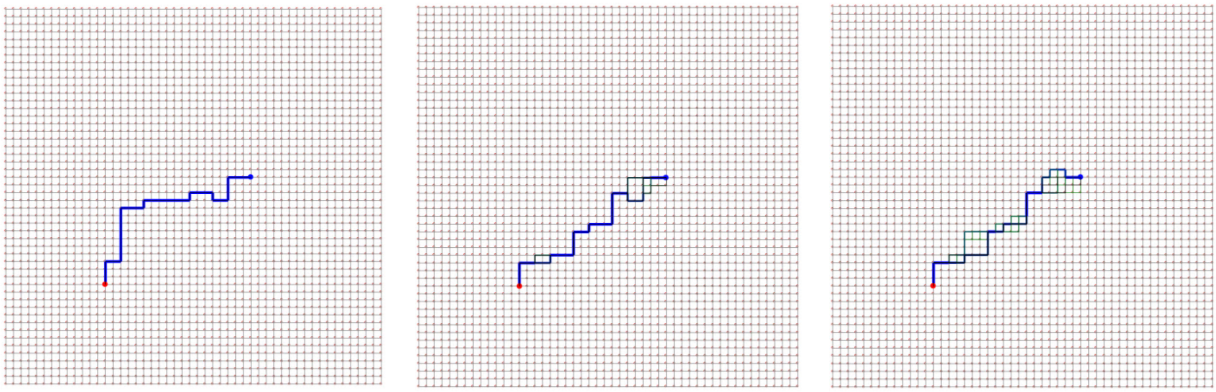



**FIGURE 4 Map display module in GeoRouting.Net**



(a) $\gamma = 0$, 39 links (b) $\gamma = 5$, 50 links (c) $\gamma = 10$, 82 links

**FIGURE 5 Effects of delay levels on reliable paths**

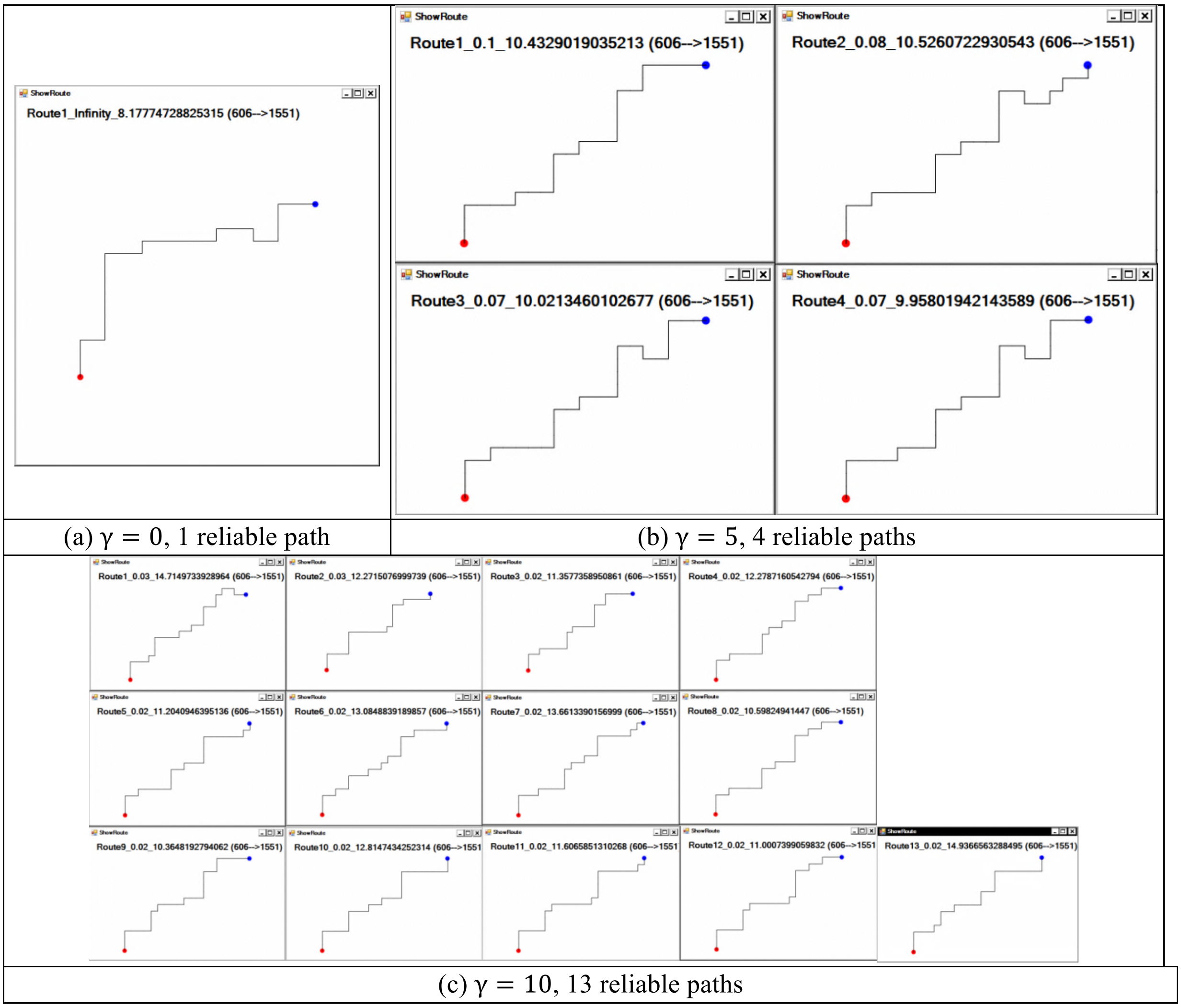



**FIGURE 6 K reliable paths generated with different delay levels**

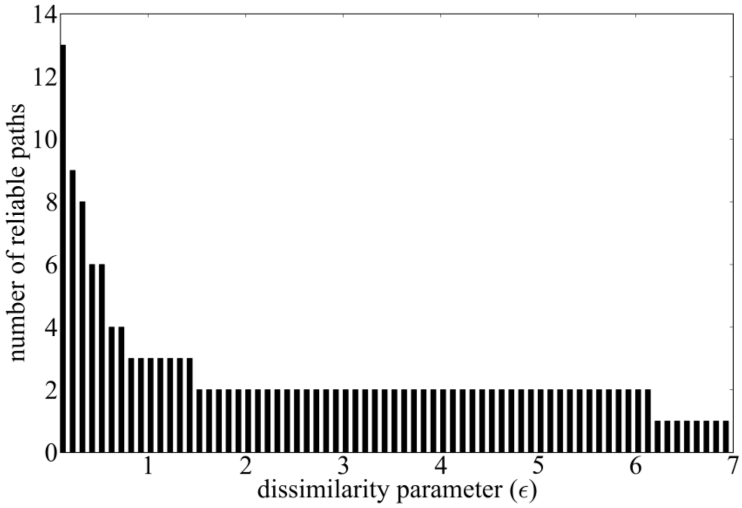



**FIGURE 7 Effects of ϵ on the number of reliable paths (interval = 0.1)**

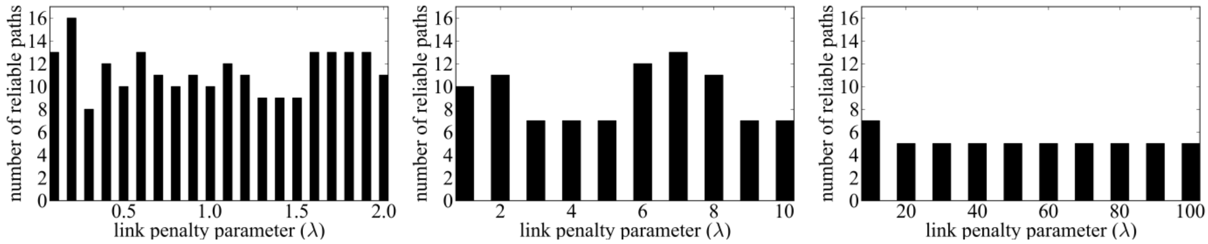



**FIGURE 8 Effects of link penalty parameter λ**