https://dx.doi.org/10.1016/j.trb.2016.09.013

Finding the K Reliable Shortest Paths under Travel Time Uncertainty

Bi Yu Chen[a*], Qingquan Li[b,a] and William H. K. Lam[c]

[a]State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan 430079, China; [b]Shenzhen Key Laboratory of Spatial Smart Sensing and Services, Shenzhen University, Shenzhen, China; [c]Department of Civil and Environmental Engineering, The Hong Kong Polytechnic University, Hong Kong

Abstract

This paper investigates the problem of finding the K reliable shortest paths (KRSP) in stochastic networks under travel time uncertainty. The KRSP problem extends the classical K loopless shortest paths problem to the stochastic networks by explicitly considering travel time reliability. In this study, a deviation path approach is established for finding K α-reliable paths in stochastic networks. A deviation path algorithm is proposed to exactly solve the KRSP problem in large-scale networks. The A* technique is introduced to further improve the KRSP finding performance. A case study using real traffic information is performed to validate the proposed algorithm. The results indicate that the proposed algorithm can determine KRSP under various travel time reliability values within reasonable computational times. The introduced A* technique can significantly improve KRSP finding performance.

Key words: K reliable shortest paths problem, travel time uncertainty, reliability

1. Introduction

The problem of finding the shortest path in road networks has been widely studied (Dijkstra, 1959; Bast et al., 2007; Geisberger et al., 2012; Zhou et al., 2014; Li et al., 2015), due to its broad application in transportation science and other fields (e.g., route guidance systems, traffic simulations, or logistics optimizations). As a natural generalization, the problem of finding K shortest paths has also received considerable attention in the literature (Hoffman and Pavley, 1959; Yen, 1971; Eppstein, 1998; Martins and Pascoal, 2003; Hershberger et al., 2007; Vanhove and Fack, 2012). Given an integer $K \geq 1$, the K shortest paths problem is intended to successively find the shortest path, the second shortest path, etc., until the Kth shortest path between an origin and a destination (O-D) pair. Provision of multiple alterative

[*] To cite this paper: Chen, B.Y., Li, Q.Q. and Lam, W.H.K., 2016, Finding the k reliable shortest paths under travel time uncertainty. Transportation Research Part B, 94, pp. 189-203.

paths is a common requirement for route guidance systems, and widely used for solving optimization problems with complex constraints and/or multiple objectives. Typical applications of the K shortest paths problem are summarized in Eppstein (1998).

In the classical K shortest paths problem, link travel times are assumed to be deterministic. However, link travel times in road networks are widely recognized by transportation practitioners and researchers to be highly stochastic, due to demand fluctuation and capacity degradation (Lam et al., 2008; Yang and Zhou, 2014; Chen et al., 2013a; Miller-Hooks and Mahmassani, 2000; Fu and Rilett, 1998). Many empirical studies have shown that, faced with travel time uncertainty, travelers tend to become risk-averse when making activity-travel decisions (Taylor 2013; Carrion and Levinson, 2012; Noland and Polak, 2002). Clearly, large travel time variations may cause undesirable late arrivals and impose a high penalty (e.g., missed flights). Therefore, travelers' concerns about travel time reliability should be incorporated into the shortest path problem, as well as the K shortest paths problem.

In recent years, much research effort has been devoted to investigating reliable shortest path problems for finding the optimal path by explicitly considering travel time reliability. Several efficient solution algorithms have been developed for solving the reliable shortest path problems in large-scale networks with uncertain travel times (Nie and Wu, 2009; Ji et al., 2011; Xing and Zhou, 2011; Chen et al., 2012; Chen et al., 2014a; Khani and Boyles, 2015; Wu, 2015). However, to the best of our knowledge, the problem of finding K reliable shortest paths (KRSP) has not received attention in the literature.

To fill the gap, this paper precisely addresses the KRSP problem. The rest of the paper is organized as follows. Section 2 provides a brief literature review related to the KRSP problem. Section 3 presents the formulation of the KRSP problem. Section 4 introduces the proposed solution algorithms for solving the KRSP problem. Section 5 reports a case study using real traffic information. Section 6 presents the conclusions and recommendations for further study.

2. Literature review

This section briefly reviews the related algorithms for solving the classical K shortest path problems and the reliable shortest path problems, to provide necessary background to the KRSP problem.

Hoffman and Pavley (1959) first investigated the K shortest paths problem. Since then, numerous algorithms have been developed for solving two variants of the K shortest paths problem. In the first variant, the K shortest paths are allowed to contain cycles (i.e., non-simple paths). For finding K non-simple paths, the best known algorithm is by Eppstein

(1998) with worst case complexity $O(|A|+|N|Log|N|+K)$, where $|A|$ and $|N|$ are the number of links and nodes in the network, respectively. In the second variant, the K shortest paths are not allowed to have cycles (called simple or loopless paths). Compared to its non-simple counterpart, this variant of finding the K loopless paths is significantly harder. Yen's algorithm (Yen, 1971) is acknowledged as the best algorithm with lowest worst-case complexity $O\big(K|N|(|A|+|N|Log|N|)\big)$. This algorithm is built on the deviation path concept established by Clarke et al. (1963), which is to find the K^{th} loopless shortest path from all paths deviating from the first K-1 loopless shortest paths. A complete survey of existing K shortest paths algorithms is out of the scope of this paper; interested readers can refer to Eppstein (1998) and Hershberger et al. (2007).

Reliable shortest path problems have been intensively studied recently. The reliable shortest path problems can be roughly classified into the most reliable path problem (Frank, 1969) and the α-reliable path problem (Chen and Ji, 2005). The most reliable path problem is to find the optimal path by maximizing the travel time reliability for a pre-determined travel time budget, while the α-reliable path problem is find the optimal path by minimizing the travel time budget for a pre-determined travel time reliability. Chen et al. (2013b) noted that these two problems apply to different route guidance scenarios. The α-reliable path problem is commonly used in pre-trip planning, where travelers have freedom to choose their departure times; whereas the most reliable path problem is used for in-vehicle routing with fixed travel time budget.

Due to their non-linear and non-additive problem structure, reliable shortest path problems cannot be solved by classical deterministic shortest path algorithms (e.g., Dijkstra's algorithm). Several effective solution algorithms have been developed to exactly solve reliable shortest path problems. Mirchandani (1976) presented a recursive algorithm to determine the most reliable path. However, the algorithm requires path enumeration and can only be applied to very small networks. Based on the first order stochastic dominance (FSD) condition, Nie and Wu (2009) proposed a multi-criteria label-correcting algorithm exactly solve the most reliable path problem and α-reliable path problem. To efficiently solve the α-reliable path problem, Chen et al. (2013b) established Mean-Budget dominance condition to reduce the number of generated FSD non-dominated paths. A multi-criteria A* algorithm was then developed for efficiently finding the α-reliable path in large-scale road networks. This multi-criteria A* algorithm was further extended to consider limited spatial correlations in road networks (Chen et al., 2012; Chen et al., 2016a). To find the α-reliable path with complete spatial correlations, Xing and Zhou (2013) proposed a Lagrangian substitution algorithm using a sample-based formulation. Zeng et al. (2015) extended the sample-based formulation by representing complete spatial correlations using a Cholesky decomposition.

Along the line of previous work (Yen, 1971; Chen et al., 2013b), this paper investigates the KRSP problem for finding K α-reliable paths in stochastic networks. Since cyclic paths are generally not relevant for road users in most transportation applications, only loopless α-reliable paths are considered. This paper is a first step towards investigating the KRSP problem with the following specific contributions:

First of all, the KRSP problem is formulated in stochastic networks based on the α-reliable path concept. The formulated KRSP problem extends the classical K loopless shortest paths problem (Yen, 1971) to stochastic networks by explicitly considering travel time reliability. The classical K shortest paths problem can be regarded as a special case of the KRSP problem under the risk-neutral scenario. Such KRSP problem can be very useful for route guidance systems and many optimization problems with complex constraints or multiple objectives in stochastic networks.

Secondly, the deviation path concept is established for finding K α-reliable paths in stochastic networks. The deviation path concept (Clarke et al., 1963) has proven effective for finding K shortest paths with linear and additive problem structure. This deviation path concept is extended in this study for finding K α-reliable paths with non-linear and non-additive problem structure. The established deviation path concept provides an effective way to determine the K^{th} α-reliable path from deviation paths of the K-1 α-reliable paths.

Thirdly, exact solution algorithms are proposed to solve the KRSP problem. A deviation path algorithm (called DP-LS) is proposed for exactly finding the K α-reliable paths in large-scale road networks. The multi-criteria label-setting algorithm (Chen et al., 2013b) is modified to solve the sub-problem of finding deviation paths of a α-reliable path. The optimality and complexity of the proposed DP-LS algorithm are rigidly proved and analyzed. Based on the A* technique, two solution algorithms (called DP-A*-Eu and DP-A*-LET) are proposed to further improve KRSP finding performance in large-scale networks.

Finally, a comprehensive case study using real traffic information is performed to demonstrate the applicability of the KRSP problem. The results indicate that the proposed DP-LS algorithm can determine KRSP under various travel time reliability values, in large scale networks with reasonable computational times. The introduced A* technique can significantly improve KRSP finding performance in large-scale road networks.

3. The K reliable shortest paths problem statement

Consider a directed network, $G(N, A)$, consisting of a set of nodes, N, and a set of links, A. Each link $a \in A$ has a tail node, a head node, and a random link travel time. Link travel times may be represented as normal, log-normal, gamma, Burr, or other distribution types (Kaparias et al., 2008; Susilawati et al., 2013; Chen et al., 2016b). A list of notations used in this paper can be found in Appendix.

Let nodes $o \in N$ and $d \in N$ represent an O-D pair. A path p^u is said to be a feasible path if it is a loopless path between the O-D pair without cycles. Given the path p^u comprising 1 nodes $(o = n_1^u, ..., n_1^u = d)$ and $1-1$ links $(a_1^u, ..., a_{1-1}^u)$, its path travel time, T^u, can be calculated by the sum of link travel times along the path,

$$T^u = \sum_{i=1}^{1-1} T_i^u \ , \tag{1}$$

where T_i^u is the travel time distribution of the i^{th} link, a_i^u, along path p^u. Clearly, the path travel time T^u is also a random variable with mean and standard deviation denoted by t^u and σ^u, respectively. To achieve α probability of on-time arrival using path p^u, the required travel time budget can be expressed as the inverse cumulative distribution function (CDF) of T^u at the α confidence level, denoted by $\Phi_{T^u}^{-1}(\alpha)$. This on-time arrival probability (i.e., α parameter) is also referred as travel time reliability (Bell and Iida, 1997). $\alpha \in (0,1)$ reflects travelers' risk taking attitude towards being late, where $\alpha > 0.5$, $\alpha = 0.5$, and $\alpha < 0.5$ represent risk-averse, risk-neutral, and risk-seeking attitudes, respectively. α may be pre-determined in the pre-trip planning scenario based on the trip purpose(s) and social characteristics.

To simplify the problem and present essential ideas, it is assumed that path travel times follow normal distributions and link travel times are mutually independent. The normal distribution assumption is common in stochastic shortest path problems (Zeng et al., 2015; Chang et al., 2005). Using real traffic data, Chen et al. (2016) empirically found that normal distributions can approximate path travel time distributions well, achieving 98.3% and 94.9% of approximation accuracy at 10th and 90th percentiles, respectively. The assumption of independent link travel time distributions can be relaxed using a two-level hierarchical network model (Chen et al., 2012) or Cholesky decomposition (Zeng et al., 2015). Using these two assumptions, the mean and standard deviation of path travel time can be expressed as

$$t^u = \sum_{i=1}^{1-1} t_i^u \ , \tag{2}$$

and

$$\sigma^u = \sqrt{\sum_{i=1}^{1-1} (\sigma_i^u)^2}\,, \tag{3}$$

where t_i^u and σ_i^u are the mean and standard deviation of link travel time T_i^u, respectively. Thus, the required travel time budget can be expressed as

$$\Phi_{T^u}^{-1}(\alpha) = t^u + z_\alpha \sigma^u\,, \tag{4}$$

where z_α is the inverse CDF of the standard normal distribution at confidence level α, and can be obtained from the standard normal table or calculated by numerical approximation.

Let Q be the solution space (or path set) consisting of all feasible paths in the network, and $Q^j = Q - \{p^1, ..., p^{j-1}\}$ be the solution space consisting of all feasible paths excluding the first *j*-1 α-reliable paths. The *j*th α-reliable path (p^j) is defined as the path with minimum travel time budget in Q^j (i.e., satisfying $\Phi_{T^j}^{-1}(\alpha) \le \Phi_{T^u}^{-1}(\alpha)\,, \forall p^u \in Q^j$). Then, the KRSP problem can be defined as below:

Definition 1. (K reliable shortest paths problem) Given integer $K \ge 1$ and travel time reliability $\alpha \in (0,1)$, the KRSP problem is to find K α-reliable paths, denoted by $\{p^1, ..., p^K\}$, satisfying

(1) $\Phi_{T^j}^{-1}(\alpha) \le \Phi_{T^{j+1}}^{-1}(\alpha)$, $\forall j \in (1, ..., K-1)$, and

(2) $\Phi_{T^K}^{-1}(\alpha) \le \Phi_{T^u}^{-1}(\alpha)$, $\forall p^u \in Q - \{p^1, ..., p^K\}$.

The KRSP problem is the generation of α-reliable path problem (Chen and Ji, 2005). When $K = 1$, the KRSP problem is equivalent to the α-reliable path problem that minimizes travel time budget (i.e., $\Phi_{T^u}^{-1}(\alpha)$) required to achieve a given travel time reliability. When $K > 1$, the KRSP problem is to find the α-reliable path, the second α-reliable path, etc., until the *K*th α-reliable path between the O-D pair.

The KRSP problem also extends the classical K shortest paths problem to stochastic networks by considering various travel time reliability concerns (i.e., $\forall \alpha \in (0,1)$). The classical K shortest paths problem can be regarded as a special case of the KRSP problem under the risk-neutral scenario (i.e., $\alpha = 0.5$), where only mean travel time is considered and the α-reliable paths become the classical shortest paths.

4. Solution algorithms

This section presents the proposed solution algorithms for solving the KRSP problem. Section 4.1 establishes the deviation path concept for the α-reliable path in stochastic networks, and Section 4.2 proposes a deviation path algorithm to solve the KRSP problem. Section 4.3 proposes two solution algorithms based on the A* technique to further improve KRSP finding performance.

4.1. Deviation path concept for the α-reliable paths

The deviation path approach has proven effective for solving the classical the K loopless shortest paths problem in the deterministic networks. This approach was originally established by Clarke et al. (1963) based on the Bellman's Principle of Optimality (Bellman, 1958), which states that the sub-path of shortest path is the shortest path itself. However, the Bellman's Principle of Optimality does not hold for the α-reliable paths with non-linear and non-additive problem structure. By relaxing such Bellman's Principle of Optimality, this study extends the deviation path concept for the α-reliable paths using a perspective of solution space decomposition.

Figure 1 illustrates the deviation path concept of the first α-reliable path through a simple example. The first α-reliable path, $p^1 = \{1,2,3,4\}$ with four nodes, has a set $\bar{D}^1$ of three deviation paths $\{\bar{p}_1^1,..,\bar{p}_i^1,...,\bar{p}_3^1\}$. Each deviation path, $\bar{p}_i^1 \in \bar{D}^1$, coincides with path p^1 for the first i nodes, then deviates to a different node and finally reaches the destination. Along deviation path $\bar{p}_i^1$ (e.g., $\bar{p}_3^1$), the i^{th} node is the deviation node, n_i^1 (e.g., node 3); the sub-path from the first node to the deviation node is the root path, $\bar{r}_i^1 = \{n_1^1,..,n_i^1\}$ (e.g., $\bar{r}_3^1 = \{1,2,3\}$); the remaining sub-path from the deviation node to the destination is the spur path, $\bar{s}_i^1$; and the deviation path is the concatenation of the root and spur paths, $\bar{p}_i^1 = \bar{r}_i^1 \oplus \bar{s}_i^1$.

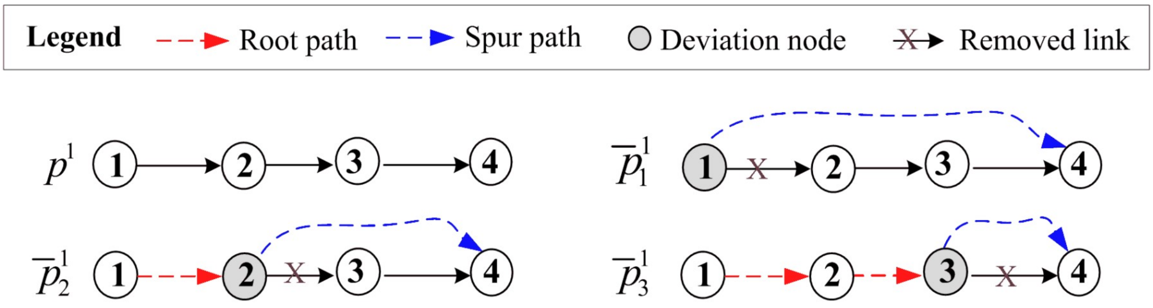


Figure 1. All possible deviation paths for the first α-reliable path

To define the i^{th} deviation path $\bar{p}_i^1 \in \bar{D}^1$, a network $\bar{G}_i^1$ can be constructed by removing the i^{th} link, $a_i^1 \in p^1$ (i.e., $a_i^1 = (n_i^1, n_{i+1}^1)$), from the original network G. Let $\bar{Q}_i^1$ be the solution space consisting of all feasible paths in $\bar{G}_i^1$ passing through the root path $\bar{r}_i^1$. Then, $\bar{p}_i^1$ is defined as the path satisfying $\Phi_{\bar{T}_i^1}^{-1}(\alpha) \le \Phi_{\bar{T}_u^1}^{-1}(\alpha)$, $\forall \bar{p}_u^1 \in \bar{Q}_i^1$. Since the i^{th} link $a_i^1 \in p^1$ is removed in $\bar{G}_i^1$, $\bar{p}_i^1$ can be guaranteed different from the first α-reliable path p^1.

Given the first α-reliable path $p^1 = \{n_1^1,...,n_i^1,...,n_1^1\}$ with $1-1$ links, there is $1-1$ deviation paths $\{\bar{p}_1^1,..,\bar{p}_i^1,...,\bar{p}_{1-1}^1\} \in \bar{D}^1$. Let Q be the solution space consisting of all feasible paths in the original network G. We have following proposition about the decomposition of solution space Q:

Proposition 1. The solution space Q can be partitioned into 1 mutually independent sub-spaces, $Q = \bar{Q}_1^1 \cup ... \cup \bar{Q}_i^1 \cup .... \cup \bar{Q}_{1-1}^1 \cup \{p^1\}$.

Proof. Since $\bar{Q}_i^1$ consists of all feasible paths passing through the root path $\bar{r}_i^1 = \{n_1^1,..,n_i^1\}$ and excluding i^{th} link $a_i^1 = (n_i^1, n_{i+1}^1) \in p^1$, we have $p^1 \notin \bar{Q}_i^1$, $\forall i \in [1, 1-1]$. Because $\bar{Q}_{i+1}^1$ comprising all feasible paths and passing through the root path $\bar{r}_{i+1}^1 = \{n_1^1,..,n_i^1,n_{i+1}^1\}$, $\bar{Q}_i^1$ and $\bar{Q}_{i+1}^1$ are mutually independent. Therefore, $\bar{Q}_1^1,....,\bar{Q}_i^1,.....,\bar{Q}_{1-1}^1,\{p^1\}$ are 1 mutually independent sub-spaces. We then prove $Q = \bar{Q}_1^1 \cup ... \cup \bar{Q}_i^1 \cup .... \cup \bar{Q}_{1-1}^1 \cup \{p^1\}$ as following.

Let Q_{1-1}^1 be the solution space that consists of all feasible paths in the original network G passing through the root path $\bar{r}_{1-1}^1 = \{n_1^1,..,n_{1-1}^1\}$. Because $\bar{Q}_{1-1}^1$ consists of all feasible paths passing through the root path $\bar{r}_{1-1}^1 = \{n_1^1,..,n_{1-1}^1\}$ and excluding the $1-1^{\text{th}}$ link $a_{1-1}^1 = (n_{1-1}^1, n_1^1)$, we have $\bar{Q}_{1-1}^1 \cup \{p^1\} = Q_{1-1}^1$. Similarly, since Q_{1-1}^1 ($\bar{Q}_{1-2}^1$) consists of all feasible paths, passing through the root path $\bar{r}_{1-2}^1 = \{n_1^1,..,n_{1-2}^1\}$ and (excluding) the $1-2$ $^{\text{th}}$ link $a_{1-2}^1 = (n_{1-2}^1, n_{1-1}^1)$, we have $\bar{Q}_{1-2}^1 \cup Q_{1-1}^1 = Q_{1-2}^1$. Therefore, we have $\bar{Q}_1^1 \cup ... \cup \bar{Q}_i^1 \cup .... \cup \bar{Q}_{1-1}^1 \cup \{p^1\} = \bar{Q}_1^1 \cup ... \cup \bar{Q}_i^1 \cup .... \bar{Q}_{1-2}^1 \cup Q_{1-2}^1 = \bar{Q}_1^1 \cup Q_1^1 = Q$.□

Based on the above proposition, the second α-reliable path, $p^2 \in Q^2 = Q - \{p^1\}$, is within one of sub-spaces, $\bar{Q}_1^1,...,\bar{Q}_{1-1}^1$. Obviously, we can prove that p^2 is one of deviation paths $\{\bar{p}_1^1,..,\bar{p}_i^1,...,\bar{p}_{1-1}^1\} \in \bar{D}^1$ as following:

Proposition 2. The second α-reliable path p^2 can be determined as the deviation path in $\bar{D}^1$ with minimum travel time budget.

Proof. According to Proposition 1, we have $p^2 \in Q^2 = Q - \{p^1\} = \bar{Q}_1^1 \cup ... \cup \bar{Q}_{1-1}^1$. Since deviation path $\bar{p}_i^1 \in \bar{D}^1$ satisfies $\Phi^{-1}_{\bar{T}_i^1}(\alpha) \le \Phi^{-1}_{\bar{T}_u^1}(\alpha)$, $\forall \bar{p}_u^1 \in \bar{Q}_i^1$, $\forall i \in [1, 1-1]$, we have $p^2 \in \bar{D}^1$. Therefore, p^2 is the deviation path in $\bar{D}^1$ with minimum travel time budget. □

The deviation path concept is further extended for the *j*th α-reliable path ($j \ge 2$). In this case, the first *j* α-reliable paths have been already determined. Unlike the first α-reliable path case, the first deviation node of the *j*th α-reliable path may not be its first node. The first deviation node n_m^j is defined as the last node of the longest sub-path coinciding with determined *j*-1 α-reliable paths. Then, given the *j*th α-reliable path $p^j = \{n_1^j, ..., n_m^j, ..., n_1^j\}$ with $1-1$ links, there is only $1-m$ deviation paths $\{\bar{p}_m^j, ..., \bar{p}_{1-1}^j\} \in \bar{D}^j$. Figure 2 shows a simple example with *j*=3. As can be seen, path $\{1, 2\}$ is the longest sub-path of p^3 coinciding with p^1; the first deviation node of p^3 can be identified as node 2; and only two deviation paths are obtained as $\{\bar{p}_2^3, \bar{p}_3^3\} \in \bar{D}^3$. The deviation path $\bar{p}_1^3$ would not be calculated for node 1 before the first deviation node, since it is identical to deviation path $\bar{p}_1^1$ of p^1.

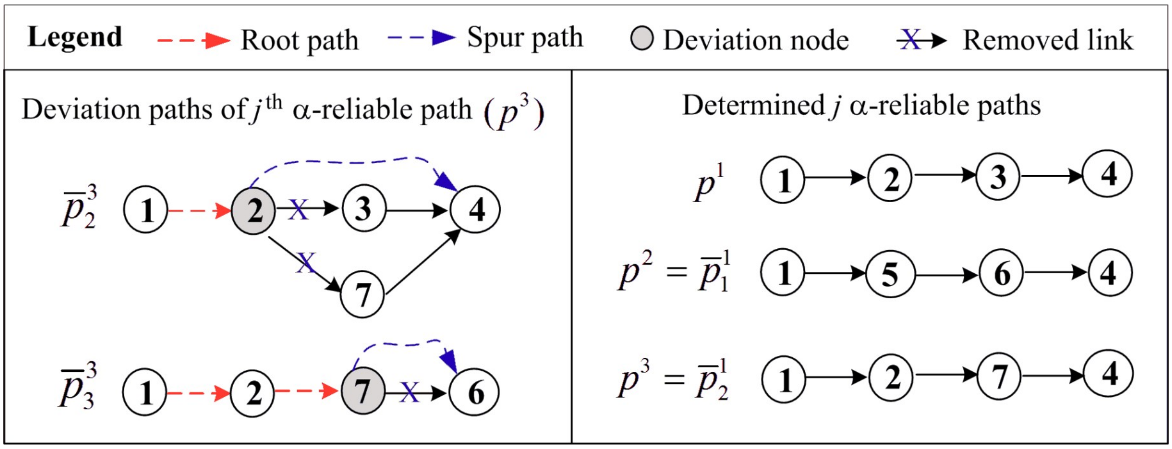


Figure 2. All possible deviation paths for the *j*th α-reliable path

To determine the *i*th deviation path (e.g., $\bar{p}_2^3$) of the *j*th α-reliable path, a network $\bar{G}_i^j$ can be constructed by removing the *i*th link $a_i^j \in p^j$ (e.g., $a_2^3 = (2, 7)$) from the original network G. Its root path $\bar{r}_i^j = \{n_1^j, ..., n_i^j\}$ is the sub-path of p^j from the first to the *i*th node. Then, a coincidence check is required to determine whether the root path $\bar{r}_i^j$ coinciding with any path amongst previous determined *j*-1 α-reliable paths. The *i*th link of all coincided paths (e.g., $a_2^1 = (2, 3) \in p^1$) should also be removed from the network $\bar{G}_i^j$. In this way, the deviation path $\bar{p}_i^j$ is guaranteed to be different from determined *j* α-reliable paths. Let $\bar{Q}_i^j$ be the solution space that consists of all feasible paths in $\bar{G}_i^j$ passing through the root path $\bar{r}_i^j$. The deviation path $\overline{p}_i^j$ is defined as the path satisfying $\Phi^{-1}_{\overline{T}_i^j}(\alpha) \le \Phi^{-1}_{\overline{T}_u^j}(\alpha), \forall \overline{p}_u^j \in \overline{Q}_i^j$.

The coincidence check is required only for determining the first deviation path $\overline{p}_m^j$, whose root path $\overline{r}_m^j$ is the longest sub-path coinciding with any determined *j*-1 α-reliable path. This coincidence check is not required for determining other deviation paths ($\overline{p}_{m+1}^j,...,\overline{p}_{1-1}^j$), because their root paths do not coincide with any path in determined *j*-1 α-reliable paths.

Without loss of generality, the *j*th α-reliable path p^j itself is assumed to be a deviation path, $\overline{p}_m^y$, of the *y*th α-reliable path p^y, $y \in [1, j-1]$. The root path $\overline{r}_m^y = \overline{r}_m^j = \{n_1^j,...,n_m^j\}$ essentially is the longest sub-path coinciding with p^j. Let G_m^y be the network for calculating the deviation path $\overline{p}_m^y$ (i.e., p^j), and $\overline{Q}_m^y$ be the solution space consisting of all feasible paths in $\overline{G}_m^y$ passing through the root path $\overline{r}_m^y$. Let $A^{\%6}_m$ be set of removed links in $\overline{G}_m^y$. To determine the *m*th deviation path (i.e., $\overline{p}_m^j$) of the *j*th α-reliable path p^j, a network $\overline{G}_m^j$ can be constructed by removing the *m*th link $a_m^j \in p^j$ from the network $\overline{G}_m^y$ (i.e., $A^{\%6}_m = A^{\%6}_m \cup \{a_m^j\}$). We have following proposition about the decomposition of solution space $\overline{Q}_m^y$:

Proposition 3. The solution space $\overline{Q}_m^y$ can be partitioned into $1-m+1$ mutually independent sub-spaces, $\overline{Q}_m^y = \overline{Q}_m^j \cup .... \cup \overline{Q}_{1-1}^j \cup \{p^j\}$.

Proof. As $\overline{Q}_i^j$ consists of all feasible paths passing through the root path $\overline{r}_i^j = \{n_1^1,..,n_i^j\}$ and excluding *i*th link $a_i^j = (n_i^j, n_{i+1}^j)$, we have $p^j \notin \overline{Q}_i^j$, $\forall i \in [m, 1-1]$. Because $\overline{Q}_{i+1}^j$ comprises all feasible paths passing through the root path $\overline{r}_{i+1}^j = \{n_1^j,..,n_i^j, n_{i+1}^j\}$, $\overline{Q}_i^j$ and $\overline{Q}_{i+1}^j$ are mutually independent. Therefore, $\overline{Q}_m^j,...,\overline{Q}_{1-1}^j,\{p^j\}$ are $1-m+1$ mutually independent sub-spaces. We then prove $\overline{Q}_m^y = \overline{Q}_m^j \cup .... \cup \overline{Q}_{1-1}^j \cup \{p^j\}$ as following.

Let Q_{1-1}^j be the solution space that consists of all feasible paths in the original network G passing through the root path $\overline{r}_{1-1}^j = \{n_1^j,..,n_{1-1}^j\}$. Because $\overline{Q}_{1-1}^j$ consists of all feasible paths passing through the root path $\overline{r}_{1-1}^j = \{n_1^j,..,n_{1-1}^j\}$ and excluding the $1-1$ th link $a_{1-1}^j = (\overline{n}_{1-1}^j, n_1^j)$, we have $Q_{1-1}^j = \overline{Q}_{1-1}^j \cup \{p^j\}$. Similarly, since Q_{1-1}^j ($\overline{Q}_{1-2}^j$) consists of all feasible paths passing through the root path $\overline{r}_{1-2}^j = \{n_1^j,..,n_{1-2}^j\}$ and (excluding) the $1-2$th link $a_{1-2}^j = (n_{1-2}^j, n_{1-1}^j)$, we have $Q_{1-2}^j = \overline{Q}_{1-2}^j \cup Q_{1-1}^j$. Therefore, we have $\overline{Q}_m^j \cup .... \cup \overline{Q}_{1-1}^j \cup \{p^j\} = \overline{Q}_m^j \cup .... \cup Q_{1-2}^j = \overline{Q}_m^j \cup Q_{m+1}^j$. Because Q_{m+1}^j is the solution space comprising all feasible paths passing through the root path $\bar{r}^j_m$ and link a^j_m, and $\bar{Q}^j_m$ is the solution space comprising all feasible paths passing through the root path $\bar{r}^j_m$ and excluding all links in $\tilde{A}^j_m = \tilde{A}^y_m \cup \{a^j_m\}$, we have $\bar{Q}^j_m \cup Q^j_{m+1} = \bar{Q}^y_m$ (i.e., the solution space consisting of all feasible paths passing through the root path $\bar{r}^j_m$ and excluding all link in $\tilde{A}^y_m$).□

Using the above proposition, the j+1th α-reliable path is within the solution space $Q^{j+1} = Q^j - \{p^j\} = (Q^j - \bar{Q}^y_m) \cup (\bar{Q}^j_m \cup .... \cup \bar{Q}^j_{l-1})$. Compared with Q^j, Q^{j+1} replaces $\bar{Q}^y_m$ with $l - m$ sub-spaces $\bar{Q}^j_m \cup .... \cup \bar{Q}^j_{l-1}$. Let $C^{j+1} = \bar{D}^1 \cup ... \cup \bar{D}^j - \{p^2, ..., p^j\}$ be the set of all deviation paths excluding the second to jth α-reliable paths. It can be proved that the j+1th α-reliable path p^{j+1} is the one of the deviation path in C^{j+1} as below.

Proposition 4. The j+1th α-reliable path p^{j+1} can be determined as the deviation path in $C^{j+1} = \bar{D}^1 \cup ... \cup \bar{D}^j - \{p^2, ..., p^j\}$ with minimum travel time budget.

Proof. According to Proposition 1, we have $Q^2 = Q - \{p^1\} = \bar{Q}^1_1 \cup .... \cup \bar{Q}^1_{l-1}$. Based on the Proposition 2, the second α-reliable path satisfies $p^2 = \bar{p}^1_i \in \bar{D}^1$, $\exists i \in [1, l-1]$. According to Proposition 3, we have $\bar{Q}^1_i = \bar{Q}^2_m \cup .... \cup \bar{Q}^2_{l-1} \cup \{p^2\}$ and thus $Q^3 = Q^2 - \{p^2\} = (Q^2 - \bar{Q}^1_i) \cup (\bar{Q}^2_m \cup .... \cup \bar{Q}^2_{l-1})$. Therefore, the third α-reliable path p^3 can be determined as the deviation path in $(\bar{D}^1 - \{p^2\}) \cup (\bar{D}^2) = \bar{D}^1 \cup \bar{D}^2 - \{p^2\} = C^3$ with minimum travel time budget.

Similarly, it can be proved that the j+1th α-reliable path is one of deviation paths in C^{j+1} as follows. Suppose that the jth α-reliable path p^j is a deviation path $\bar{p}^y_m$ of the yth α-reliable path p^y, $\exists y \in [1, j-1]$. According to Proposition 3, we have $Q^{j+1} = Q^j - \{p^j\} = (Q^j - \bar{Q}^y_m) \cup (\bar{Q}^j_m \cup .... \cup \bar{Q}^j_{l-1})$. Therefore, j+1th α-reliable path p^{j+1} can be determined as the path in $(C^j - \{p^j\}) \cup (\bar{D}^j) = \bar{D}^1 \cup ... \cup \bar{D}^j - \{p^2, ..., p^j\} = C^{j+1}$ with minimum travel time budget. □

Using Propositions 2 and 4, the K α-reliable paths can be determined subsequently by calculating deviation paths of the K-1th α-reliable paths. This provides an effective way for finding the K α-reliable paths in stochastic networks. In the next section, deviation path algorithms will be developed for exactly solving the KRSP problem.

4.2. Deviation path algorithm

In this section, a deviation path algorithm (named DP-LS) is proposed for finding the K α-reliable paths in stochastic networks.

The steps of the DP-LS algorithm is described below. Initially, the first α-reliable path, p^1, is determined. In this study, the multi-criteria label-setting algorithm (called *RSPP-LS*) (Chen et al., 2013b) is adopted due to its computational efficiency. In each iteration, the *j*+1th α-reliable path, p^{j+1}, is determined. To fulfill this task, all deviation paths of the *j*th α-reliable path (p^j) are calculated using the CalculateDeviations(p^j) procedure. The calculated deviation paths (stored in $\bar{D}^j$) are then added into a sorted list, C, as candidate paths and ordered by their travel time budgets. The candidate path at the top of C (with minimum travel time budget) is the *j*+1th α-reliable path, p^{j+1}, and is removed from C. The sorted list C can be implemented using efficient priority queue structures, such as the F-heap data structure (Fredman and Tarjan 1987). The algorithm continues until the *K*th α-reliable path is found.

In the DP-LS algorithm, the CalculateDeviations procedure determines all deviations of the *j*th α-reliable path, p^j. The procedure determines the first deviation node (n_m^j) of p^j as the last node of the longest sub-path coinciding with determined *j*-1 α-reliable paths $\{p^1,...,p^{j-1}\}$. Then, a coincidence check is carried out to check whether the root path $\bar{r}_m^j = \{n_1^j,...,n_m^j\}$ coincides with any determined α-reliable path, p^y. If so, the *i*th link, $a_i^y = \{n_i^y, n_{i+1}^y\}$, of any coincided path p^y is removed from the network G to the removed link set $\tilde{A}$. This is to guarantee that $\bar{p}_m^j$ deviates from previous determined *j*-1 α-reliable paths. To facilitate the first deviation node search and coincidence check, the deviation tree structure (Roddity and Zwick, 2005) can be used. By storing and searching *j*-1 α-reliable paths in this tree structure, the first deviation node (n_m^j) and coincided paths ($\forall p^y$) can be efficiently determined.

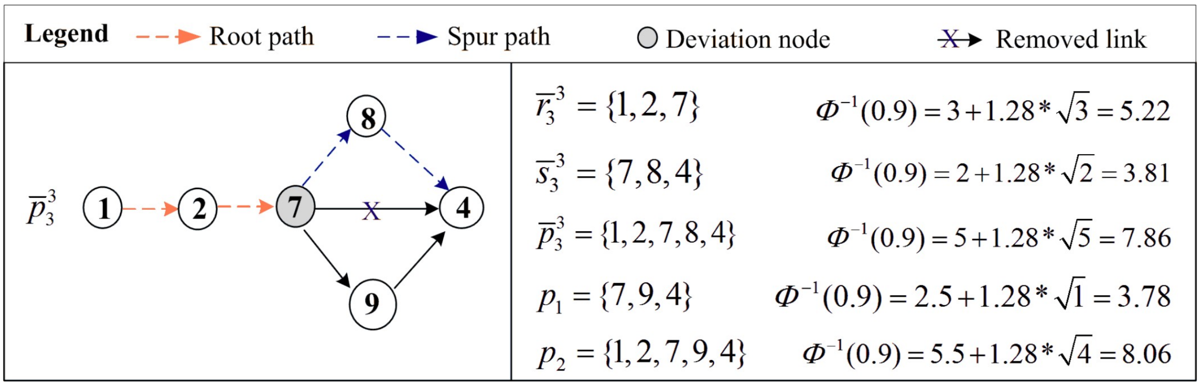
The procedure then iteratively calculates the deviation path, $\bar{p}_i^j$, for each node n_i^j ($\forall i = m,...,1-1$). The root path $\bar{r}_i^j = \{n_1^j,...,n_i^j\}$ is set as the first node to node n_i^j. The *i*th link $a_i^j \in p_i^j$ (i.e., $a_i^j = \{n_i^j, n_{i+1}^j\}$) is then removed from the original network G to the link set $\tilde{A}$ (so as to construct network $\bar{G}_i^j$). This is to guarantee that deviation path $\bar{p}_i^j$ is different from p^j. Subsequently, the deviation path $\bar{p}_i^j$ is found in the network G by using the $MRSPLS(G, \bar{r}_i^j, d, \alpha)$ algorithm, which is a modification of the original multi-criteria label-setting algorithm (Chen et al., 2013b). After determination of $\bar{p}_i^j$, the network G is restored by adding all links in $\tilde{A}$. The procedure continues until all deviation paths, $\{\bar{p}_m^j,...,\bar{p}_{1-1}^j\}$, are calculated and stored in $\bar{D}^j$.

```
**Algorithm:** DP-LS
**Inputs**: O-D pair, travel time reliability $\alpha$, and number of α-reliable paths K
**Returns**: Sorted collection L of K α-reliable paths
Find the first α-reliable path $p^1 := RSPLS(G, o, d, \alpha)$.
Set candidate path set $C := \phi$ (paths are sorted by their travel time budget).
For $j$ from 1 to $K$-1
    Calculate deviation path set $\bar{D}^j :=$ CalculateDeviations($p^j$).
    Set $C := C \cup \bar{D}^j$.
    Set $p^{j+1}$ as the candidate path at the top of $C$.
    Set $L := L \cup \{p^{j+1}\}$ and $C := C - \{p^{j+1}\}$.
End For
Return $L$.

**Procedure:** CalculateDeviations
**Inputs**: The $j^{th}$ α-reliable path $p^j = (n_1^j, ..., n_m^j, ..., n_1^j)$
**Returns**: The deviation path set $\bar{D}^j$ of $p^j$
Set $\bar{D}^j := \phi$.
Determine the first deviation node $n_m^j$ of $p^j$.
Set link set $\tilde{A} := \phi$.
For each path $p^y$ ($y = 1, ..., j-1$)
    If $\bar{r}_m^j \in p^y$, then add $a_m^y \in p^y$ into $\tilde{A}$.
End For
For $i$ from $m$ to $1-1$($1$ is the number of nodes in $p^j$)
    Set root path $\bar{r}_i^j := \{n_1^j, ..., n_i^j\}$.
    Add $a_i^j \in p^j$ into $\tilde{A}$.
    Remove links in $\tilde{A}$ from the network $G$.
    Find deviation path $\bar{p}_i^j := MRSPLS(G, \bar{r}_i^j, d, \alpha)$.
    Set $\bar{D}^j := \bar{D}^j \cup \{\bar{p}_i^j\}$.
    Restore network $G$ by adding links in $\tilde{A}$, and set $\tilde{A} := \phi$.
End for
Return $\bar{D}^j$.
```

Compared with the classical K shortest path algorithm (Yen, 1971), the proposed DP-LS algorithm uses a distinct way of calculating deviation path $\bar{p}_i^j$. In Yen's algorithm, every node in the root path $\bar{r}_i^j = \{n_1^j, ..., n_i^j\}$ is firstly removed from the network $\bar{G}_i^j$; then the spur path $\bar{s}_i^j$ from the deviation node n_i^j to the destination n_{1-1}^j is determined by using Dijkstra's algorithm in the network $\bar{G}_i^j$; and finally the deviation path $\bar{p}_i^j$ is obtained by concatenating $\bar{r}_i^j$ and $\bar{s}_i^j$. However, in the proposed DP-LS algorithm, the deviation path $\bar{p}_i^j$ cannot be calculated by determining the spur path $\bar{s}_i^j$ in the network $\bar{G}_i^j$ as the path with the minimum travel time budget from n_i^j to n_{1-1}^j, and then combining it with the root path $\bar{r}_i^j$. This is because the α-reliable path does not satisfy Bellman's principle of optimality. In this case, the actual spur path $\bar{s}_i^j$ may not be the α-reliable path with minimum travel time budget from n_i^j to n_{1-1}^j.

Figure 3 illustrates this issue through a simple example. As can be seen, the travel time budget of deviation path $\bar{p}_3^3 = \{1,2,7,8,4\}$ is not equal to the summation of travel time budget of root path $\bar{r}_3^3 = \{1,2,7\}$ and spur path $\bar{s}_3^3 = \{7,8,4\}$ (i.e., $7.86 < 5.22 + 3.81 = 9.03$). This is due to the non-linear and non-additive problem structure of α-reliable path. It can be observed clearly that the actual spur path $\bar{s}_3^3 = \{7,8,4\}$ is not identical to the α-reliable path $p_1 = \{7,9,4\}$ with minimum travel time budget from node 8 to node 5. The actual deviation path $\bar{p}_3^3 = \{1,2,7,8,4\}$ can be mis-identified as $p_2 = \{1,2,7,9,4\}$ by simply combining root path $\bar{r}_3^3$ and p_1.



$$\bar{r}_3^3 = \{1,2,7\} \qquad \Phi^{-1}(0.9) = 3 + 1.28 * \sqrt{3} = 5.22$$

$$\bar{s}_3^3 = \{7,8,4\} \qquad \Phi^{-1}(0.9) = 2 + 1.28 * \sqrt{2} = 3.81$$

$$\bar{p}_3^3 = \{1,2,7,8,4\} \qquad \Phi^{-1}(0.9) = 5 + 1.28 * \sqrt{5} = 7.86$$

$$p_1 = \{7,9,4\} \qquad \Phi^{-1}(0.9) = 2.5 + 1.28 * \sqrt{1} = 3.78$$

$$p_2 = \{1,2,7,9,4\} \qquad \Phi^{-1}(0.9) = 5.5 + 1.28 * \sqrt{4} = 8.06$$

Figure 3. Non-additive issue in calculating deviation paths of the α-reliable path

To determine the deviation path $\bar{p}_i^j$ in stochastic networks, the original multi-criteria label-setting algorithm (i.e., *RSPP-LS*) (Chen et al., 2013b) is modified in this study. In the *RSPP-LS* algorithm, a path p^{rr} from the origin to itself is generated and added into the priority queue in the initialization step (see Page 134 in Chen et al., 2013b). In the modified algorithm (i.e., $MRSPLS$), the root path $\overline{r}_i^{\,j}$ is generated and added into the priority queue instead of p^{rr}. In this way, all paths generated during the $MRSPLS$ algorithm are guaranteed to pass through $\overline{r}_i^{\,j}$. When the $MRSPLS$ algorithm terminates, the deviation path $\overline{p}_i^j$ can be determined as the path between the O-D pair with minimum travel time budget, passing through the root path $\overline{r}_i^{\,j}$. Otherwise, no deviation path can be obtained. Since cycles are not permitted in the multi-criteria label-setting algorithm, no cycle occurs in the determined deviation path $\overline{p}_i^j$.

The optimality of the proposed DP-LS algorithm is proved as follows.

Proposition 5. The proposed DP-LS algorithm can determine optimal K α-reliable paths $\{p^1,...,p^K\}$ when the algorithm terminates.

Proof. Since the *RSPP-LS* algorithm provides an exact solution for solving the α-reliable path problem (Propositions 8, Chen et al., (2013b)), the first α-reliable path can be exactly determined. From Proposition 2, the second α-reliable path can be determined as one of deviation paths of the first α-reliable path. In the first iteration, all deviation paths of the first α-reliable path are determined by the $\text{CalculateDeviations}$ procedure and maintained in the sorted list, C. Therefore, the second α-reliable path is the top path of C with minimum travel time budget. The *j*+1th α-reliable path can then be determined at the *j*th iteration ($j = 2,...,K-1$) as below.

From Proposition 4, the *j*+1th α-reliable path, p^{j+1}, is the deviation path in $C^{j+1} = \overline{D}^1 \cup ... \cup \overline{D}^j - \{p^2,...,p^j\}$ with minimum travel time budget. In the algorithm, all deviation paths in C^{j+1} are iteratively determined. Therefore, the *j*+1th α-reliable path p^{j+1} can be exactly determined. □

The complexity of DP-LS algorithm is analyzed. In the worst case, each *j*th α-reliable path can have $|N|$ deviation paths, where $|N|$ is the number of nodes in the network. The *MRSPLS* algorithm runs in $O(|A||P|+|N|Log|N|)$ when the F-heap data structure is used, where $|A|$ is the number of links in the network and $|P|$ is the maximum number of non-dominated paths at the nodes. Therefore, the proposed deviation path algorithm runs in $O\big(K|N|(|A||P|+|N|Log|N|)\big)$. Theoretically, $|P|$ grows exponentially with the network size. In practice, $|P|$ is much smaller than the maximum possible size especially for sparse networks used in various transportation applications (Chen et al., 2013b; Nie and Wu, 2009).

4.3. The A* technique for improving computational performance

Due to numerous deviation path searches in the DP-LS algorithm, the KRSP computational performance highly depends on *MRSPLS* algorithm. In this study, the multi-criteria A* technique (Chen et al., 2013b) is adopted to improve the *MRSPLS* algorithm. The multi-criteria A* technique introduces a heuristic function, $h(i)$, to estimate the lower bound of travel time budget from node i to the destination. Using this A* technique, the path searching performance can be improved by assigning a high priority to those nodes closer to the destination.

The effectiveness of A* technique relies on the quality of $h(i)$. The better estimation of travel time budget from node i to the destination, the higher computational improvement can be achieved. The algorithm can guarantee to obtain the optimal solution if the heuristic function $h(i)$ is admissible. As discussed in Chen et al. (2013b), the Euclidean heuristic function $h(i)=d^{id}/v_{\max}$ is an admissible function for any risk-taking scenario (i.e., $\forall\alpha\in(0,1)$), where d^{id} is the Euclidean distance from node i to the destination d, and $v_{\max}$ is the maximum travel speed (or maximum design speed) in the network. Since Euclidean distance between any two nodes can be directly calculated based on their coordinates, the Euclidean heuristic function can be easily incorporated into the *MRSPLS* algorithm. To distinguish it from the DP-LS algorithm, the algorithm using the Euclidean heuristic function is called DP-A*-Eu algorithm.

For risk-neutral (i.e., $\alpha=0.5$) and risk-averse (i.e., $\alpha>0.5$) scenarios, a better admissible function $h(i)=t_*^{id}$ can be utilized, where t_*^{id} is the least expected travel time from node i to the destination d. Using this expected travel time heuristic function, a solution algorithm, DP-A*-LET, is developed to efficiently solve the KRSP problem. The DP-A*-LET algorithm needs to perform an additional all-to-one backward Dijkstra's algorithm before the first α-reliable path searching. This Dijkstra's algorithm generates a backward shortest path tree rooted at the destination, which maintains the least expected travel time t_*^{id} from each node i to the destination.

It should be noted that the DP-A*-LET algorithm cannot be used for risk-seeking (i.e., $\alpha<0.5$) scenarios, because the mean travel time heuristic function is not admissible in these scenarios. Nevertheless, both DP-LS and DP-A*-Eu algorithms can be used for all risk-taking scenarios (i.e., $\forall\alpha\in(0,1)$).

5. Case study

5.1. Numerical example

A case study was performed to demonstrate the applicability of the proposed algorithms for solving the KRSP problem. The real road network of Wuhan city, China was used, which consisted of 19,306 nodes and 46,757 links (see Figure 4). Using floating car data during the morning peak period (7–10 a.m.), link travel time distributions in the Wuhan network were estimated and calibrated into either lognormal, gamma, or normal distributions. Detailed information on the Wuhan network has been documented in Chen et al. (2016b).

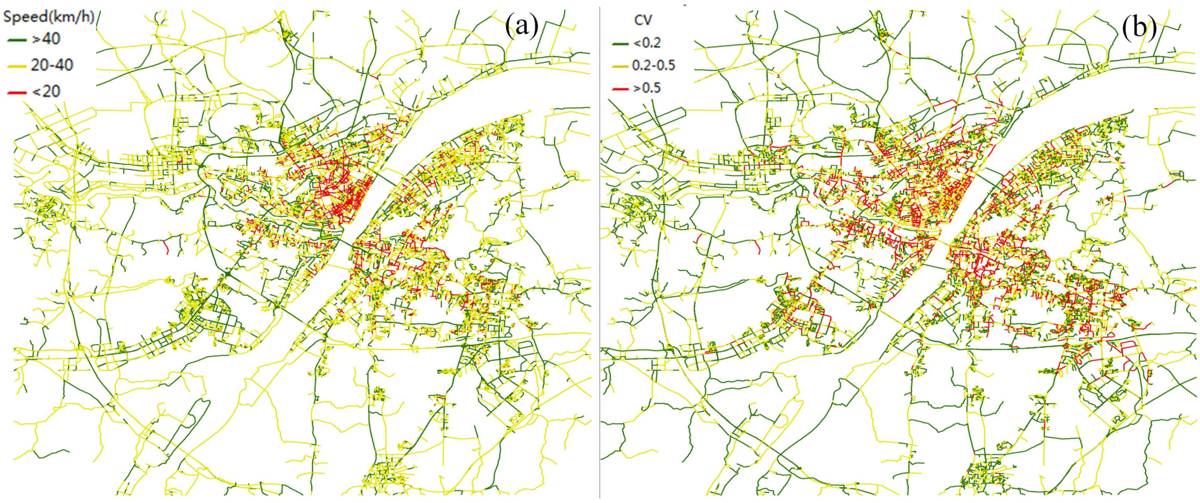


Figure 4. Traffic condition of the Wuhan network: (a) mean link speed and (b) coefficient of variation of link travel time distributions

As shown in Figure 5, one O-D pair was selected to demonstrate the KRSP finding results. The K parameter was set as 3 and the travel time reliability parameter, α, was set as 0.9. The first, second and third α-reliable paths are shown in the figure using orange, dark blue, and light blue, respectively. The travel time distribution information of three determined α-reliable paths are given in Table 1. These three α- reliable paths had very closed travel time budgets, ($\Phi_{T^1}^{-1}(0.9) = 1754.14 < \Phi_{T^2}^{-1}(0.9) = 1756.67 < \Phi_{T^3}^{-1}(0.9) = 1758.97$). They are three paths with least travel time budgets amongst all feasible paths for the $\alpha = 0.9$ scenario. Therefore, the KRSP problem generalizes the previous α-reliable path problem (Chen and Ji, 2005), which only determines the first α-reliable path between the O-D pair.

When using the multi-criteria optimization technique to solve the α-reliable path problem (Nie and Wu, 2009; Chen et al., 2013b), several non-dominated paths can be generated between the O-D pair. It is noted that the j^{th} α-reliable paths ($j = 2, \ldots, K$) may not be non-dominated paths between the O-D pair. For example, the second α-reliable path is M-B dominated by the first α-reliable path, since $t^1 = 1661.25 < t^2 = 1669.75$ and $\Phi_{T^1}^{-1}(0.9) = 1754.14 < \Phi_{T^2}^{-1}(0.9) = 1756.67$. Thus, the j^{th} α-reliable paths ($j = 2,...,K$) may not be included in the set of generated non-dominated paths when solving the α-reliable path problem.

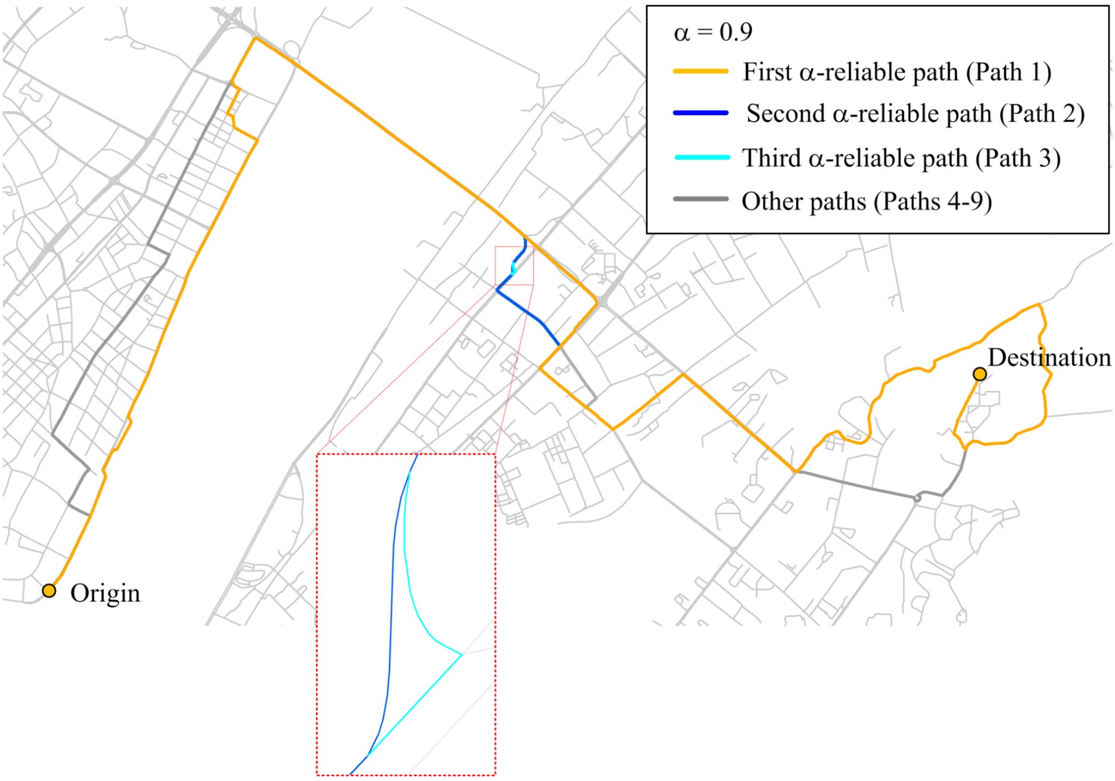

α = 0.9
First α-reliable path (Path 1)
Second α-reliable path (Path 2)
Third α-reliable path (Path 3)
Other paths (Paths 4-9)
Destination
Origin

Figure 5. The K reliable shortest paths finding results under risk-averse scenario

Table 1. The K reliable shortest path finding results (in seconds)

Travel time reliability (α)	α-reliable paths	Mean	Standard deviation	Travel time budget
0.9	1st α-reliable path (Path 1)	1661.25	72.51	1754.14
	2nd α-reliable path (Path 2)	1669.75	67.85	1756.67
	3rd α-reliable path (Path 3)	1672.12	67.79	1758.97
0.5	1st α-reliable path (Path 4)	1653.66	88.81	1653.66
	2nd α-reliable path (Path 5)	1657.52	88.83	1657.52
	3rd α-reliable path (Path 6)	1657.74	93.26	1657.74
0.1	1st α-reliable path (Path 7)	1663.82	107.25	1526.43
	2nd α-reliable path (Path 8)	1659.74	103.40	1527.28
	3rd α-reliable path (Path 9)	1667.68	107.27	1530.26

Using the same O-D pair, the KRSP finding results were examined under different travel time reliability (α). Figure 6 shows the KRSP finding results when $\alpha = 0.1$. Compared to the risk-averse counterpart (Figure 5), travelers in this risk-seeking scenario tend to take the risk of being late by choosing paths with larger travel time variations. Figure 7 shows the determined three α-reliable paths when $\alpha = 0.5$. In this risk-neutral scenario, travelers search for their optimal paths only based on mean travel times, and ignore travel time variations. Three generated α-reliable paths are paths with the first, second and third least expected travel times. In this case, the KRSP problem reduces to the classical K shortest paths problem (Yen, 1971). Therefore, the proposed KRSP problem extends the classical K shortest paths problem by incorporating travelers' travel time reliability concerns (i.e., $\forall \alpha \in (0,1)$). The classical K shortest paths problem can be regarded as a special case of the KRSP problem under risk-neural scenario.

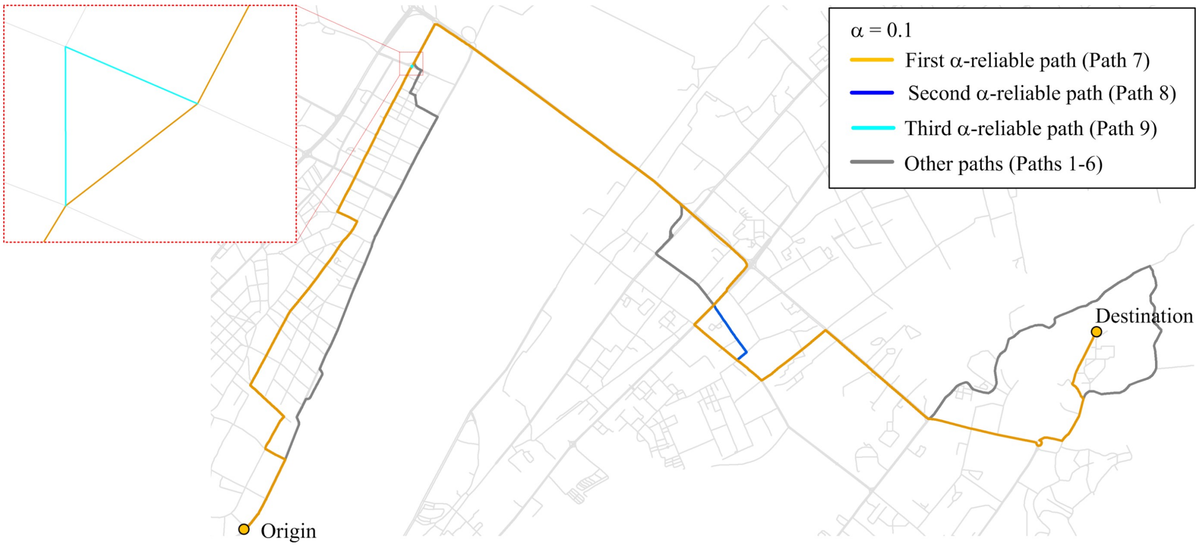

α = 0.1
First α-reliable path (Path 7)
Second α-reliable path (Path 8)
Third α-reliable path (Path 9)
Other paths (Paths 1-6)
Destination
Origin

Figure 6. The K reliable shortest paths finding results under risk-seeking scenario

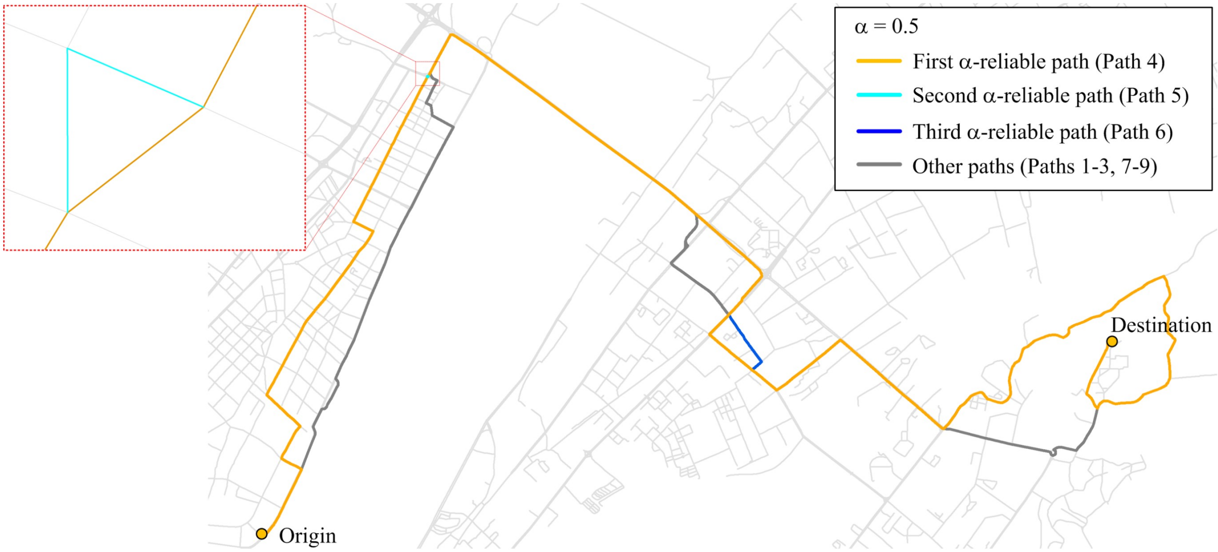

α = 0.5
First α-reliable path (Path 4)
Second α-reliable path (Path 5)
Third α-reliable path (Path 6)
Other paths (Paths 1-3, 7-9)
Destination
Origin

Figure 7. The K reliable shortest paths finding results under risk-neutral scenario

5.2. Computational performance

The computational performance of the proposed algorithms (including DP-LS, DP-A*-Eu and DP-A*-LET) were examined on several large-scale networks. All algorithms were coded in Visual C# programming language, with the F-heap data structure (Fredman and Tarjan, 1987) implemented as a sorted list C for ordering generated deviation paths. The same F-heap data structure was used in the *MRSPLS* algorithm for searching deviation paths. All computational experiments were conducted on a MacBook Air laptop with 4 core 2.0 GHz CPU (only one core was used) and 8 GB RAM, running the Windows 7 operating system.

Six testing networks were adopted, including three road and three grid networks. Apart from the Wuhan network, the other five networks and their associated link travel time distributions were obtained from Chen et al. (2013b). The basic characteristics the testing networks are shown in Table 2. For each testing network, 100 O-D pairs were randomly generated, and all testing algorithms provided the same set of K α reliable paths. Computational times were the average of 100 runs using the generated O-D pairs.

Table 2. Basic characteristics of testing networks

Road networks			Grid networks		
Network	Number of nodes	Number of links	Network	Number of nodes	Number of links
Hong Kong RTIS	1,367	3,655	G1 (40*50)	2,000	7,820
Chicago regional	12,982	39,018	G2 (50*100)	5,000	19,700
Wuhan network	19,306	46,757	G3(100*100)	10,000	39,600

Table 3 shows the algorithms' computational performance under the risk-averse scenario, $\alpha = 0.9$. The K value was set as 100. As shown in the table, all three algorithms employed the same large number of deviation path searches when solving the KRSP problem. The computational performance of three algorithms depends highly on that of deviation path searches. For example, the DP-LS algorithm consumed 70.656 seconds to find the first 100 α-reliable paths in the Wuhan network. The first α-reliable path search consumed only 0.0994 seconds, however the searches of 2893 deviation paths required 70.551 seconds, accounting for over 99.8% of total computational time. The search of the first deviation node using the deviation tree structure was very efficient and only consumed less than 0.006 seconds in each test.

Both DP-A*-Eu and DP-A*-LET algorithms run significantly faster than the DP-LS algorithm. For example, to find the first 100 α-reliable paths in the Wuhan network, the DP-A*-Eu algorithm required 47.145 seconds, which was 49.9% faster than the DP-LS algorithm. This computational improvement was achieved due to the utilization of the Euclidean heuristic function (i.e., $h(i) = d^{id} / v_{\max}$) in the DP-A*-Eu algorithm for speeding up the deviation path searches. Compared to DP-A*-Eu, the DP-A*-LET algorithm further improved deviation path searches by almost 240% in the Wuhan network by utilizing the expected travel time heuristic function (i.e., $h(i) = t_*^{id}$). In view of this computational improvement, the additional 0.146 seconds consumed by the DP-A*-LET algorithm for constructing backward shortest path tree is reasonable and worth implementing. Compared to the DP-LS algorithm, the DP-A*-LET algorithm can improve the KRSP finding performance more than 5 times in the Wuhan network and 20 times in the G3 network. Therefore, the introduced A* technique can significantly improve the KRSP finding performance.

Computational performance is strongly affected by network size. For example, the DP-LS algorithm required 1.124 seconds to determine the first 100 α-reliable paths in the Hong Kong RTIS network with 1367 nodes, whereas for the Wuhan network (approximately 13 times as large, 19306 nodes) the computational time sharply increased 61 times to 70.656 seconds. This was to be expected, since the computational performance of the *MRSPLS* algorithm used for deviation path searches is degraded with network size (Chen et al., 2013b). In addition, the K α-reliable paths tend to have more nodes in the larger networks, leading to more deviation path searches required. For example, the number of deviation path searches in the Wuhan network was 2.3 times that of the Hong Kong RTIS network (see Table 3).

Table 3. Computational time of three algorithms under risk-averse scenario (in seconds)

Network	Hong	Chicago	Wuhan	G1	G2	G3

	Kong	regional	network			
Algorithm	RTIS					
DP-LS	*1.124*	76.535	**70.656**	1.915	11.406	41.888
FPS	0.0027	0.114	0.0994	0.0068	0.0284	0.0894
DPS	1.119	76.416	70.551	1.905	11.374	41.794
DP-A*-Eu	0.925	24.922	**47.145**	0.792	4.667	13.284
FPS	0.0023	0.065	0.0725	0.004	0.020	0.0472
DPS	0.920	24.855	47.066	0.785	4.644	13.232
DP-A*-LET	0.237	4.814	**14.054**	0.187	0.704	2.048
BTC	0.0035	0.0788	0.146	0.0078	0.0251	0.0736
FPS	0.0002	0.0165	0.0044	0.0005	0.0034	0.0077
DPS	0.232	4.717	13.897	0.176	0.672	1.960
DPS Number	*1266*	2887	**2893**	1192	1771	2435

BTC: Backward tree construction; FPS: First α-reliable path search; DPS: Deviation path search

The KRSP finding performance of the proposed algorithms was also examined under the risk-seeking scenario. Because the DP-A*-LET algorithm cannot be used for the risk-seeking scenarios, only the computational performance of DP-LS and DP-A*-Eu algorithms were tested, as shown in Table 4. The α value was set to 0.1 and the K value was set to 100. The DP-A*-Eu algorithm effectively improves the computational performance of the KRSP search under risk-seeking scenarios. For example, the DP-A*-Eu algorithm was 54.1% and 204.6% faster than the DP-LS algorithm in Wuhan and Chicago regional networks, respectively.

Table 4. Computational time of two algorithms under risk-seeking scenario (in seconds)

Network	Hong	Chicago	Wuhan	G1	G2	G3
	Kong	regional	network			
Algorithm	RTIS					
DP-LS	1.034	*49.511*	**64.159**	1.581	8.607	31.989
FPS	0.0023	0.0614	0.068	0.0056	0.0183	0.056
DPS	1.029	49.444	64.085	1.573	8.585	31.928
DP-A*-Eu	0.821	*16.255*	**41.624**	0.610	3.242	9.773
FPS	0.0021	0.0345	0.0653	0.003	0.012	0.0288
DPS	0.817	16.216	57.553	0.604	3.226	9.739
DPS Number	1223	2872	2978	1171	1771	2474

FPS: First α-reliable path search; DPS: Deviation path search

The KRSP finding performance was examined for various α values, as shown in Figure 8 for the most efficient algorithm in the Wuhan network. Under risk-averse and risk-neutral scenarios (i.e., $0.5 \le \alpha \le 0.99$), the DP-A*-LET algorithm was adopted. It can be observed that the computational performance of the DP-A*-LET algorithm improved when α approaches 0.5. For example, the algorithm required 16.179 seconds when $\alpha = 0.99$; and only 10.752 seconds when $\alpha = 0.5$, because the KRSP problem reduces to the classical K shortest path problem. In the risk-neutral scenario, the multi-criteria A* algorithm for deviation path searches reduces to the classical A* algorithm; and only one shortest path is determined for each deviation path search. However, with increasing α, larger numbers of non-dominated paths are generated by the multi-criteria A* algorithm (Chen et al., 2013b), leading to increased computational times.

The same pattern can be observed in Figure 8 for the risk-seeking scenarios (i.e., $0.01 \le \alpha < 0.5$). Computational performance of the DP-A*-Eu algorithm improved when α approaches 0.5. For example, the DP-A*-Eu algorithm consumed 52.179 seconds when $\alpha = 0.1$, which reduced by 16.5% to 43.569 seconds when $\alpha = 0.4$. This is because less non-dominated paths are generated during the deviation path search process when α approaches 0.5.

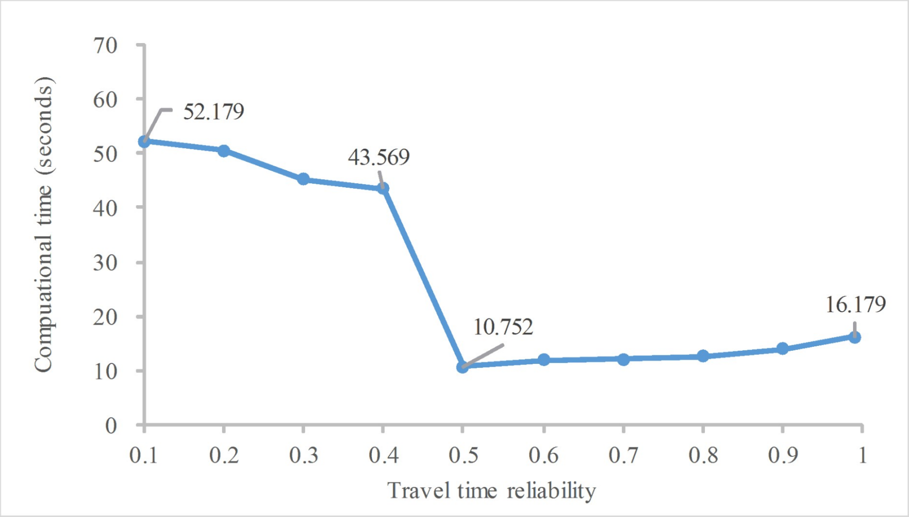


Figure 8. Computational performance for various travel time reliability values

Finally, the effects of different K values on the KRSP finding performance was investigated. Figure 9 shows the KRSP finding performance in the Wuhan network using DP-A*-LET algorithm for the risk-averse scenario of $\alpha = 0.9$. Computational times linearly increased with the K value. For example, the algorithm consumed 0.769 seconds when K=10, which increased approximately 16.5 times to 13.476 seconds when K=100. This result is expected, because more computational effort is required for deviation path searches with increased K. For example, when K increased from 10 to 100, the number of deviation path searches increased by about 5.8 times from 428 to 2926.

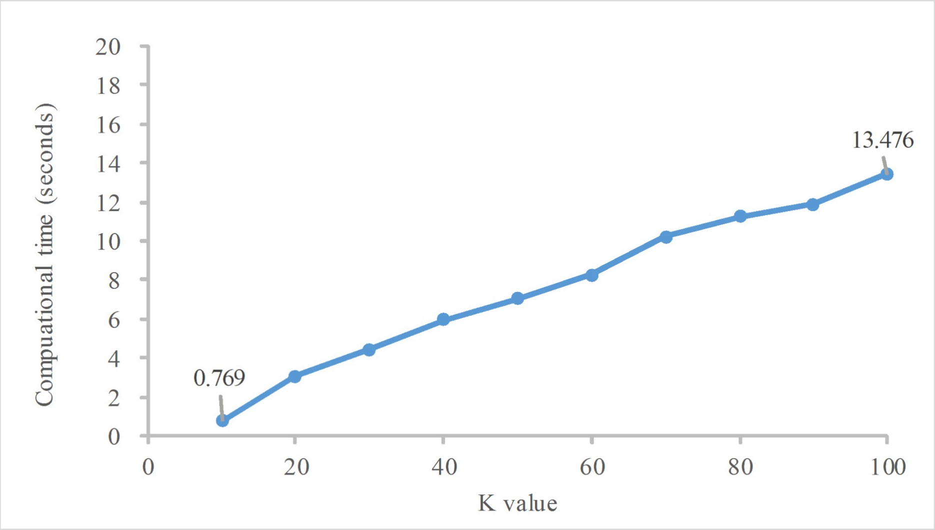


Figure 9. Computational performance for different K values

6. Conclusions and future research

This paper investigated the K reliable shortest path (KRSP) problem in stochastic networks under travel time uncertainty, which is to successively find the first α-reliable shortest path, the second α-reliable shortest path, etc., until the K^{th} α-reliable shortest path between the origin-destination (O-D) pair. The KRSP problem extends the classical K loopless shortest paths problem (Yen, 1971) to stochastic networks by explicitly considering travel time reliability. This KRSP problem can be very useful for route guidance systems and many optimization problems with complex constraints or multiple objectives in stochastic networks.

In this study, the deviation path concept for the deterministic shortest path (Yen, 1971) was established for the α-reliable paths in stochastic networks from a perspective of solution space decomposition. A deviation path algorithm (DP-LS) was proposed to exactly solve the KRSP problem. Optimality and computational complexity of the proposed DP-LS algorithm were proved and analyzed. Two solution algorithms (DP-A*-Eu and DP-A*-LET) based on the A* heuristic technique were developed to further improve the KRSP finding performance. A case study using real traffic information was performed out to demonstrate the applicability of the KRSP problem. The results indicated that the proposed solution algorithms can determine the K α-reliable shortest paths under various travel time reliability values (i.e., $\forall \alpha \in (0,1)$). The proposed DP-LS algorithm can exactly determine the K α-reliable shortest paths in large-scale networks in reasonable computational times. The proposed A* technique can significantly improve KRSP finding performance, for example the DP-A*-LET algorithm can speed up the KRSP finding performance by more than 20 times in the G3 network.

Future research interests will cover three major extensions. First, path travel times were assumed to follow normal distributions and the correlations of link travel times were omitted for simplicity. Previous studies have found that travel times could be better approximated by correlated and shifted log-normal travel times (Srinivasanet al., 2014). How to address the KRSP problem using correlated and shifted log-normal travel times requires further investigation. Second, the KRSP problem was formulated for road networks. The extension of the KRSP problem to multi-modal transportation networks will be a topic for future work. Third, the KRSP problem was formulated based on the α-reliable path concept. However, other models of reliable shortest path have been developed in different application contexts, such as expected utility theory (Wu and Nie, 2011). The extension of the KRSP problem to other reliable shortest path models will be another topic for further studies.

Acknowledgements

The authors are thankful to two anonymous referees for their comments and suggestions that improved this paper. The work described in this paper was jointly supported by research grants from the National Natural Science Foundation of China (Nos. 41231171 and 41571149) and the Research Grants Council of the Hong Kong Special Administrative Region (Project Nos. PolyU 5242/12E and 152074/14E).

References

Bast, H., Funke, S., Sanders, P. and Schultes, D., 2007, Fast routing in road networks with transit nodes. *Science*, 316, p. 566.

Bell, M.G.H. and Iida, Y., 1997, Transportation Network Analysis (New York: John Wiley & Sons).

Bellman, R., 1958, On a routing problem. *Quarterly of Applied Mathematics*, 16, pp. 87-90.

Carrion, C. and Levinson, D., 2012, Value of travel time reliability: A review of current evidence. *Transportation Research Part A*, 46, pp. 720–741.

Chang, T.S., Nozick, L.K. and Turnquist, M.A., 2005, Multiobjective path finding in stochastic dynamic networks, with application to routing hazardous materials shipments. *Transportation Science*, 39, pp. 383-399.

Chen, A. and Ji, Z.W., 2005, Path finding under uncertainty. *Journal of Advanced Transportation*, 39, pp. 19-37.

Chen, B.Y., Lam, W.H.K., Sumalee, A. and Li, Z.L., 2012, Reliable shortest path finding in stochastic networks with spatial correlated link travel times. *International Journal of Geographical Information Science*, 26, pp. 365-386.

Chen, B.Y., Lam, W.H.K., Li, Q.Q., Sumalee, A. and Yan, K., 2013a, Shortest path finding problem in stochastic time-dependent road networks with stochastic first-in-first-out property. *IEEE Transactions on Intelligent Transportation Systems*, 14, pp. 1907-1917.

Chen, B.Y., Lam, W.H.K., Sumalee, A., Li, Q.Q., Shao, H. and Fang, Z.X., 2013b, Finding reliable shortest paths in road networks under uncertainty. *Networks & Spatial Economics*, 13, pp. 123-148.

Chen, B.Y., Lam, W.H.K., Sumalee, A., Li, Q.Q. and Tam, M.L., 2014a, Reliable shortest path problems in stochastic time-dependent networks. *Journal of Intelligent Transportation Systems*, 18, pp. 177-189.

Chen, B.Y., Lam, W.H.K. and Li, Q.Q., 2016a, Efficient solution algorithm for finding spatially-dependent reliable shortest path in road networks. *Journal of Advanced Transportation*, Conditional accepted.

Chen, B.Y., Shi, C., Zhang, J., Lam, W.H.K., Li, Q.Q. and Xiang, S., 2016b, Most reliable path-finding algorithm for maximizing on-time arrival probability. *Transportmetrica B*, DOI:10.1080/21680566.2016.1169953.

Clarke, S., Krikorian, A. and Rausen, J., 1963, Computing the N best loopless paths in a network. *Journal of the Society for Industrial and Applied Mathematics*, 11, pp. 1096-1102.

Dijkstra, E.M., 1959, A note on two problems in connexion with graphs. *Numerische Mathematic*, 1, pp. 269-271.

Eppstein, D., 1998, Finding the k shortest paths. *SIAM Journal on Computing*, 28, pp. 652-673.

Frank, H., 1969, Shortest paths in probabilistic graphs. *Operations Research*, 17, pp. 583-599.

Fredman, M.L. and Tarjan, R.E., 1987, Fibonacci heaps and their uses in improved network optimization algorithms. *Journal of the ACM*, 34, pp. 596-615.

Fu, L.P. and Rilett, L.R., 1998, Expected shortest paths in dynamic and stochastic traffic networks. *Transportation Research Part B*, 32, pp. 499-516.

Geisberger, R., Sanders, P., Schultes, D. and Vetter, C., 2012, Exact routing in large road networks using contraction hierarchies. *Transportation Science*, 46, pp. 388-404.

Hershberger, J., Maxel, M. and Suri, S., 2007, Finding the k shortest simple paths: A new algorithm and its implementation. *ACM Transactions on Algorithms*, 3, p. Article 45.

Hoffman, W. and Pavley, R., 1959, A method for the solution of the Nth best path problem. *Journal of the ACM*, 6, pp. 506-514.

Ji, Z.W., Kim, Y.S. and Chen, A., 2011, Multi-objective alpha-reliable path finding in stochastic networks with correlated link costs: A simulation-based multi-objective genetic algorithm approach (SMOGA). *Expert Systems with Applications*, 38, pp. 1515-1528.

Kaparias, I., Bell, M.G.H. and Belzner, H., 2008, A new measure of travel time reliability for in-vehicle navigation systems. *Journal of Intelligent Transportation Systems*, 12, pp. 202-211.

Khani, A. and Boyles, S.D., 2015, An exact algorithm for the mean-standard deviation shortest path problem. *Transportation Research Part B*, 81, pp. 252-266.

Lam, W.H.K., Shao, H. and Sumalee, A., 2008, Modeling impacts of adverse weather conditions on a road network with uncertainties in demand and supply. *Transportation Research Part B*, 42, pp. 890-910.

Li, Q.Q., Chen, B.Y., Wang, Y. and Lam, W.H.K., 2015, A hybrid link-node approach for finding shortest paths in road networks with turn restrictions. *Transactions in GIS*, 19, pp. 915-929.

Martins, E.Q.V. and Pascoal, M.M.B., 2003, A new implementation of Yen's ranking loopless paths algorithm. *4OR*, 1, pp. 121-133.

Miller-Hooks, E.D. and Mahmassani, H.S., 2000, Least expected time paths in stochastic, time-varying transportation networks. *Transportation Science*, 34, pp. 198-215.

Nie, Y. and Wu, X., 2009, Shortest path problem considering on-time arrival probability. *Transportation Research Part B*, 43, pp. 597-613.

Noland, R.B. and Polak, J.W., 2002, Travel time variability: A review of theoretical and empirical issues. *Transport Reviews*, 22, pp. 39-54.

Srinivasan, K.K., Prakash, A.A. and Seshadri, R., 2014, Finding most reliable paths on networks with correlated and shifted log-normal travel times. *Transportation Research Part B*, 66, pp. 110-128.

Susilawati, S., Taylor, M.A.P. and Somenahalli, S., 2013, Distributions of travel time variability on urban roads. *Journal of Advanced Transportation*, 47, pp. 720-736.

Taylor, M.A.P., 2013, Travel through time: the story of research on travel time reliability. *Transportmetrica B*, 1, pp. 174-194.

Vanhove, S. and Fack, V., 2012, An effective heuristic for computing many shortest path alternatives in road networks. *International Journal of Geographical Information Science*, 26, pp. 1031-1050.

Wu, X. and Nie, Y., 2011, Modeling heterogeneous risk-taking behavior in route choice: A stochastic dominance approach. *Transportation Research Part A*, 45, pp. 896-915.

Wu, X., 2015, Study on mean-standard deviation shortest path problem in stochastic and time-dependent networks: A stochastic dominance based approach. *Transportation Research Part B*, 80, pp. 275-290.

Xing, T. and Zhou, X., 2011, Finding the most reliable path with and without link travel time correlation: A Lagrangian substitution based approach. *Transportation Research Part B*, 45, pp. 1660-1679.

Xing, T. and Zhou, X., 2013, Reformulation and solution algorithms for absolute and percentile robust shortest path problems. *IEEE Transactions on Intelligent*

Transportation Systems, 14, pp. 943 - 954.

Yang, L. and Zhou, X., 2014, Constraint reformulation and a Lagrangian relaxation-based solution algorithm for a least expected time path problem. *Transportation Research Part B*, 59, pp. 22-44.

Yen, J.Y., 1971, Finding the k shortest loopless paths in a network *Management Science*, 17, pp. 712-716.

Zeng, W., Miwa, T., Wakita, Y. and Morikawa, T., 2015, Application of Lagrangian relaxation approach to a-reliable path finding in stochastic networks with correlated link travel times. *Transportation Research Part C*, 56, pp. 309-334.

Zhou, Y., Wang, W., He, D. and Wang, Z., 2014, A fewest-turn-and-shortest path algorithm based on breadth-first search. *Geo-spatial Information Science*, 17, pp. 201-207.

Appendix. Notations

A	A set of links in the network
$\overset{\%}{A}_i^{6}$	The set of removed links for constructing $\bar{G}_i^j$
a_i^u	The ith link along path p^u
C^j	The set of deviation paths from the j-1 α-reliable paths while excluding the second to the j-1th a-reliable paths
$\bar{D}^j$	The set of deviation paths of the jth α-reliable path
d	Destination node
d^{id}	Euclidean distance from node i to the destination d
G	Original network
$\bar{G}_i^j$	A network for calculating deviation path $\bar{p}_i^j$
$h(i)$	Heuristic function used in the A* technique
N	A set of nodes in the network
n_i^u	The ith node along path p^u
o	Origin node
p^u	A feasible from the origin to the destination with no cycle
p^j	The jth a-reliable path
$\bar{p}_i^j$	The ith deviation path for the jth α-reliable path
Q	The solution space consisting of all feasible paths in the network
Q^j	The solution space consisting of all feasible paths excluding the first j-1 α-reliable paths
$\bar{r}_i^j$	The root path of deviation path $\bar{p}_i^j$
$\bar{s}_i^j$	The spur path of deviation path $\bar{p}_i^j$

T^u	Travel time distribution of path p^u
T_i^u	Travel time distribution of link a_i^u
t^u	Mean of path travel time distribution T^u
t_i^u	Mean of link travel time distribution T_i^u
t_*^{id}	The least expected travel time from origin i to destination d
z_α	The inverse of cumulative distribution function of the standard normal distribution at confidence level α
α	Travel time reliability (i.e., on-time arrival probability)
σ^u	Standard deviation of path travel time distribution T^u
σ_i^u	Standard deviation of link travel time distribution T_i^u
$\Phi_{T^u}^{-1}(\alpha)$	Travel time budget required to achieve α travel time reliability using path p^u