Reliable a Priori Shortest Path Problem with Limited Spatial and Temporal Dependencies

Chapter · August 2009

DOI: 10.1007/978-1-4419-0820-9_9

2 authors:


Yu Nie
Northwestern University


Xing Wu
Lamar University

Chapter 9

Reliable a Priori Shortest Path Problem with Limited Spatial and Temporal Dependencies

Yu (Marco) Nie and Xing Wu, Northwestern University, USA

Abstract This paper studies the problem of finding most reliable a priori shortest paths (RASP) in a stochastic and time-dependent network. Correlations are modeled by assuming the probability density functions of link traversal times to be conditional on both the time of day and link states. Such correlations are spatially limited by the Markovian property of the link states, which may be such defined to reflect congestion levels or the intensity of random disruptions. We formulate the RASP problem with the above correlation structure as a general dynamic programming problem, and show that the optimal solution is a set of non-dominated paths under the first-order stochastic dominance. Conditions are proposed to regulate the transition probabilities of link states such that Bellman's principle of optimality can be utilized. We prove that a non-dominated path should contain no cycles if random link travel times are consistent with the stochastic first-in-first-out principle. The RASP problem is solved using a non-deterministic polynomial label correcting algorithm. Approximation algorithms with polynomial complexity may be achieved when further assumptions are made to the correlation structure and to the applicability of dynamic programming. Numerical results are provided.

1. Introduction

Routing in a stochastic network has important applications in numerous science and engineering fields. In transportation, motorists and freight carriers need to account for random road travel times when making schedule and routing decisions. A variety of stochastic routing problems have been studied in this context, of which many assume that road travel times follow an independent probability distribution. However, correlations among road travel times are easy to bring to mind: incidents on freeways may cause delays on local streets; a bad snowstorm may simultaneously affect all roads in certain areas. Such a general correlation structure is difficult to fully quantify, and hence must be simplified in one way or another. Polychronopoulos and Tsitsiklis (1996) took the correlation of travel times into account through a discrete joint distribution, when formulating the shortest path problem with recourse. The recourse problem, in which the traversal time on a link becomes known and fixed upon the arrival of its tail (starting) node, is shown to be NP-hard even without correlation. Waller and Ziliaskopoulos (2002) studied a simplified recourse problem known as recourse with reset (see Provan 2003). Their model assumes that the link traversal time distribution is conditional on the state of the preceding link. It is worth noting that the temporal dependency in Waller and Ziliaskopoulos (2002) refers to recourse. i.e., travel time can be learned before traversing a link. In this paper, however, temporal dependency is defined in the sense of Hall (1986), i.e., the traversal time distribution is conditional on time. In Fan et al. (2005b), a similar limited correlation structure is employed whereas link states are restricted to be either congested or uncongested. Gao and Chabini (2006) extended the recourse problem to time-dependent networks, and focused on a simplified version in which travel times on all links become known at the beginning of each time interval. Similar to Polychronopoulos and Tsitsiklis (1996), Gao et al.'s model embedded correlations in a discrete joint distribution. Another approach to account for correlations is using covariance matrices (e.g. Sivakumar and Batta 1994; Sen et al. 2001). However, dynamic programming (DP) is generally inapplicable to these models. Instead, nonlinear or integer programming formulations were suggested.

In this research, travel time correlations are restricted only to adjacent links, along the line of Waller and Ziliaskopoulos (2002) and Fan et al. (2005b), and travel time distribution are allowed to vary over time, similar to the work of Hall (1986), Miller-Hooks and Mahmassani (2000), Fu and Rilett (1998), Fu (2001), Miller-Hooks (2001) and Miller-Hooks and Mahmassani (2003). For a probabilistic network with such a correlation structure, the present paper studies the *reliable a priori shortest path problem* (RASP), which aims to find *a priori* paths that are shortest to ensure a specified probability of on-time arrival. Similar to Frank (1969), our definition of optimality directly addresses the reliability in routing. In contrast, most aforementioned models that consider correlations (Polychronopoulos and Tsitsiklis 1996; Waller and Ziliaskopoulos 2002; Fan et al. 2005b; Gao and Chabini 2006) seek to minimize expected travel times.

We emphasize that the RASP problem targets *a priori* path generation rather than adaptive routing. The latter, which enables real-time rerouting decisions (e.g. Fan el al. 2005a; Nie and Fan 2006), is simpler because standard dynamic programming techniques can be applied. *A priori* paths are useful in many circumstances. From a modeling point of view, generating *a priori* optimal paths constitutes an important part in reliable network equilibrium and design models. From a practical point view, a motorist usually begins a trip bearing in mind a planned route, and en-route rerouting occurs only when the travel time on the pre-planned route exceeds a certain threshold. Furthermore, in reality motorists often need to make a routing decision before they arrive the next decision point (e.g. when they need to exit a congested freeway). Last but not least, an adaptive optimal policy may suggest cycling (Nie and Fan 2006), a counterintuitive guidance that motorists are unlikely to take. On the contrary, an *a priori* path may not contain cycles even in a stochastic environment.

The RASP problem studied in this paper belongs to a class of multi-criteria shortest path problems (e.g., Hansen 1979; Loui 1983; Miller-Hooks and Mahmassani 2000), which rely on a dominance relationship to obtain Pareto-optimal solutions against multiple criteria. The dominance relationship in the RASP problem is defined with respect to the cumulative distribution function (CDF) of path traversal times. Therefore, the dimension of the criterion vector is infinite in theory, albeit discretization is often required to obtain a tractable problem. The dominance defined by CDF is known as the *first-order stochastic dominance* (FSD) (e.g. Levy 1992; Muller and Stoyan 2002). While this fundamental concept has found a broad range of applications in statistics, economics, finance, etc., its application in stochastic shortest path problems has received little attention. Notable exceptions include Miller-Hooks (1997), Miller-Hooks and Mahmassani (2003), and recently Nie and Wu (2008). Miller-Hooks and Mahmassani (2003) proposed three rules to establish dominance among paths, one of which equals FSD. Label-correcting algorithms were proposed to find non-dominated paths according to each of the three rules (Miller-Hooks 1997). Nie and Wu (2008) studied a continuous formulation of a similar problem, in which the CDF of path travel time is generated recursively from a convolution integral. Unlike Miller-Hooks (1997), Nie and Wu's model defines optimality with respect to arrival rather than departure time. Various approximation schemes, some of which leads to polynomial complexity, were also discussed in Nie and Wu (2008).

The present paper extends Nie and Wu's work (2008) by introducing limited spatial correlations into the RASP problem. Specifically, the probability density function of the traversal time on a link is assumed to be conditional on the *state* of that link. The states of a link have a Markovian property. Namely, the state of present link is dependent on the state of the link traversed right before arriving at its starting node, and independent of the links traversed prior to that. The probability distribution of link traversal times is also allowed to vary over time, which provides a mechanism to account for the dynamic network behavior such as congestion effects caused by rush hour traffic. For a given destination, the optimal path at a node is determined by the following inputs: the desired arrival time, the probability of arriving on time, and the observed states on the immediate successor links. We shall show that Bellman's principle of optimality applies to the optimal paths of the correlated RASP problem, when the transition probability of link states meets some regulatory conditions. This result allows us to formulate the RASP problem using general dynamic programming (Carraway et al. 1990), and solve it using label-correcting-type algorithms. In order to establish the acyclicity of optimal paths and hence to insure the finiteness of label-correcting algorithms, a stochastic first-in-first-out (FIFO) condition is proposed.

The remainder of the paper is organized as follows. The next section describes the correlated RASP problem, gives necessary definitions and provides an illustrative example. Section 3 presents a dynamic programming formulation based on the first-order stochastic dominance and proves its validity. Section 4 discusses the solution algorithm and its complexity. Numerical results are reported in Section 5 and Section 6 concludes the paper.

2. Problem Statement

Consider a directed network $G(N, A, P)$ consisting a set of nodes N, a set of links A, a probability distribution P describing the statistics of the link traversal times. Let $|N| = n$ and $|A| = m$. The analysis period is set to $[0, T]$. Let the destination of routing be s and the desirable arrival time be aligned with the end of the analysis period T. A list of notation can be found in Appendix A.

Let x_{ij} denote the random *state* of link *ij*, and $x_{ij} \in \chi(i, j)$ which is a set of possible states on link ij. Although we do not impose any constraints on χ (other than being a finite set), in practice the state of a link may be defined according to the level of congestion (low, medium or high) or the presence of major disruptions (yes or no). In what follows, we assume that all $\chi(i, j), \forall ij \in A$ are identical, i.e., all links have the same state set χ. Accordingly, the subscript in the link state x_{ij} is dropped for simplicity whenever it does not cause confusion.

To consider limited spatial correlations, we assume that the states have a Markovian property, in the sense that, conditional on the state of the link traversed right before arriving at a node j (say link ij), the state of link jl is independent of the links traversed prior to that.

Define ρ_{ijl}^{xy} as the transition probability from state x on link ij to state y on link jl in the *forward* chain. For every link pair ij and jl, ρ_{ijl} defines a $|\chi| \times |\chi|$ transition probability matrix. We shall call ρ_{ij} a *transition mapping* for link ij, which includes a set of transition probability matrices that correspond to all $jl \in A$. Note that the above definition implies that a link state may have different probabilities in the presence of multiple predecessor links, which raises a consistency issue in real applications. Although the inconsistency may be resolved by defining transition probabilities on the "combined" states of all (instead of individual) incoming links, it is difficult to build a tractable model in this framework. A simpler remedy would be to force the transition matrices of all incoming links to be identical, which can be imposed when constructing transition probability matrices. The latter treatment is compatible with the models and algorithm considered herein.

Further restrictions are needed on the transition mapping in order to obtain a tractable model. Specifically, we only focus on transition mapping that satisfies the following regulatory conditions.

Definition 1. (*Regulated transition mapping*) A transition mapping ρ_{ij} is regulated if

$$\rho_{ijl}^{xy} \geq 0, \forall x, y \in \chi, \ \sum_{y} \rho_{ijl}^{xy} = 1, \forall jl \in A, \forall x \in \chi, \tag{1}$$

$$\rho_{ijl}^{xy} = \rho_{ijl'}^{xy}, \forall l \neq l', jl, jl' \in A, \forall x, y \ . \tag{2}$$

The first condition ensures that ρ_{ijl} defines a valid discrete probability distribution. The second condition, forcing all $\rho_{ijl}, \forall jl \in A$ to be identical, is imposed for the modeling convenience, as explained in the next section.[1]

The state of the link traversed right before arriving at a node j, as well as the state of link jl, affects the traversal time on link jl. Let f_{ijx}^{t} denote the probability density function of c_{ij} when the state on link ij is x, and the arriving time at node i is t. Let π_{kx}^{rs} be the traversal time on path k^{rs} when the state of the first link on k^{rs} is x. Finally, let $u_{kx}^{rs}(\cdot)$ denote the cumulative distribution function of π_{kx}^{rs}. Note that in our setting each time budget b identifies a unique departure time instant $T-b$. Thus $u_{ky}^{js}(b)$ represents the probability of on-time arrival using path k^{js} when the state on the first link is y and leaving node j (or equivalently, arriving at node j since no waiting at node is allowed) at $T-b$. Let ij denote the preceding link of path k^{js} (whose first link is jl), we have

$$u_{kx}^{is}(b) = \sum_{y} \rho_{ijl}^{xy} \int_{0}^{b} u_{ky}^{js}(b-w) f_{ijx}^{T-b}(w) \, dw, \forall b, \forall i \neq s, \forall x_{ij} \in \chi \ , \tag{3}$$

$$\rho_{iss'}^{x0} = 1, \forall is \in A, x \in \chi, u_{k0}^{ss'}(b) = 1, \forall b \ . \tag{4}$$

Note that equation (4) delineates boundary conditions by introducing a dummy outgoing link ss' at the destination node s with a unique deterministic state 0.

The reliable a priori shortest path problem (RASP) concerned in this paper is to find, starting from any node $i \neq s$, *a priori* paths which are shortest to ensure a specified probability of arriving the destination s on time, provided the states on all outgoing links $il \in A$. By solving RASP, one can make use of the information of the states on the adjacent downstream links (hence account for the spatial correlation) to determine the best *a priori* path which maximizes the probability of on-time arrival. However, unlike the case of adaptive routing such as in Andreatta

---

[1] We note that the second regulation condition may not be as restrictive as it sounds. First, if one link is congested, the probability of becoming congested is likely to be similar across all of its outgoing links, thanks to drivers' route choice behavior (i.e., drivers are likely to select another outgoing link if the congestion probability of one outgoing link is significantly higher, which may eventually bring the congestion probability on all outgoing links to a balance). Second and perhaps more important, transition probability only dictates the possibility of changing from one state to another state. The probability density function of traversal time associated with each state (which actually matters in decision making) is specific to each link.

and Romeo (1988), the optimal path is fixed once departed and no further observation of link states are needed. Such a routing mode occurs when travelers prefer a predetermined path and have access to local traffic information only at the origin (e.g. at home). Moreover, the dynamic behavior of link traversal times (e.g. the larger mean and variance caused by rush hour traffic) is accounted through the dependency of the probability density function on the entry time of a link.

A "brute-force" method to solve RASP is to enumerate all path-state combinations and calculate the associated path distribution functions $u_{kx}^{is}(b)$ by recursively applying equation (3). The best path for a given reliability of on-time arrival at any possible realization of link states can then be identified by examining all the distribution functions. Certainly such a method is impractical because enumerating paths is prohibitively expensive. Path enumeration can be avoided in a class of shortest problems (of which the conventional shortest problem is a well-known instance) whose optimal solutions obey Bellman's principle of optimality (Bellman 2003). Unfortunately, the RASP problem is not of this type because an extension of an optimal path is not necessarily optimal. As a result, one has to retain and examine all subpaths whose extensions may become optimal. What is essential in this process is how to identify subpaths that can be safely discarded without violating the optimality. In this sense, the RASP is a multi-criteria shortest path problem (Hansen 1979; Loui 1983), which typically relies on a non-dominance relationship to obtain Pareto-optimal solutions. As mentioned before, the non-dominance relationship in the RASP problem should be defined against the entire cumulative distribution function $u_{kx}^{is}(b), \forall b$. This is often known as first-order stochastic dominance (FSD) (Levy 1992). In order to define FSD for the RASP problem, let us first introduce the concept of *successor link state*.

Definition 2. (*Successor link (SL) state*) A successor link state w at node i is a realization of states for all links $ij \in A(i)$, where $A(i)$ is the set of outgoing links from node i.

A set of feasible successor link states at node i is denoted as W_i, which may be determined from historical data and may vary from one node to another. In any case, we note that $1 \leq |W_i| \leq |A(i)|^{|\chi|}$. Moreover, at a given $w \in W_i$, the state x of the first link on any path k^{is} is uniquely determined. Hereafter we shall denote this state as a function of w for a given path, or $x = k^{is}(w)$.

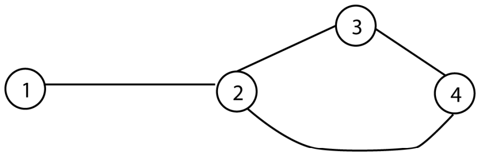
Definition 3. (*First-order stochastic dominance (FSD)* $\succ_1^w$) Path k^{rs} dominates path l^{rs} in the first order at an SL state $w \in W_r$, denoted as $k^{rs} \succ_1^w l^{rs}$, if 1) $u_{kx}^{rs}(b) \geq u_{ly}^{rs}(b), \forall b \in [0,T]$; and 2) $\exists$ at least one open interval $\Lambda \subset [0,T]$ with nonzero Lebesgue measure such that $u_{kx}^{rs}(b) > u_{ly}^{rs}(b), \forall b \in \Lambda$, where $x = k^{is}(w), y = l^{is}(w)$.

Thus, a path is *dominated* under the above definition of FSD if it is dominated at all possible SL states. In this paper, a *non-dominated* path is called FSD-admissible, as formally defined below:

Definition 4. (*FSD-admissible path*) A path l^{rs} is FSD-admissible if $\exists$ no path k^{rs} such that $k^{rs} \succ_1^w l^{rs}$ for all $w \in W_r$.

In the next section, we shall show that the RASP problem can be formulated in a general dynamic programming framework (Carraway et al. 1990) based on a non-dominance relationship aligned with FSD. Before moving to that endeavor, let us first demonstrate the concept using a small example.

Fig. 1 depicts a network with four nodes and four links where node 4 is the destination. For the demonstration purpose distributions are assumed to be discrete and depend only on the link states (not entry time). Also, there are two possible link states, 0 and 1. Note that link 2-3 has a deterministic link traversal time 0 at either state. Transition probabilities between link 1-2 and 2-3, and link 1-2 and 2-4 are specified and given in Fig. 1. For example, the data suggest that, when link 1-2 is at state 0, the probability of link 2-4 at state 0 is 0.9. Note that this transition probability mapping is not regulated because it does not meet Condition (2). Further, the feasible SL states at node 1 and 2 are $W_1 = \{\{0_{12}\},\{1_{12}\}\}$ and $W_2 = \{\{1_{23}, 1_{24}\},\{1_{23}, 0_{24}\}\}$.



Transition prob.	Link 2-3	
Link 1-2	State0	State1
State0	(0)	(1.0)
State1	(0)	(1.0)

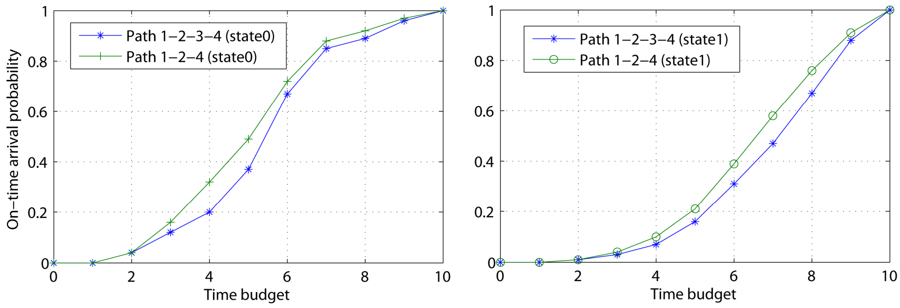
Transition prob.	Link 2-4	
Link 1-2	State0	State1
State0	(0.9)	(0.1)
State1	(0.9)	(0.1)

Probability distribution of link traversal times

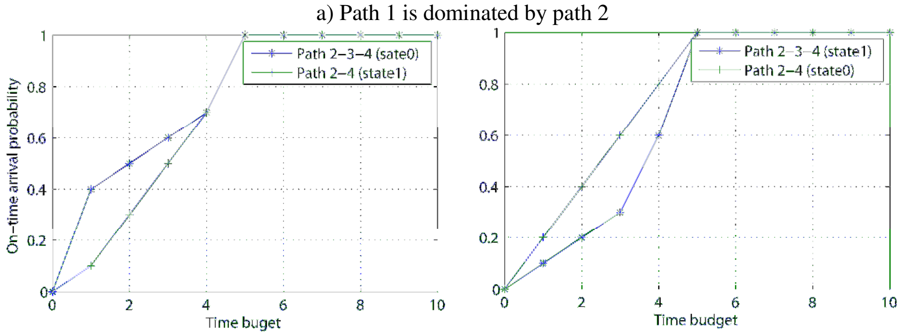
Link	1-2		2-3		2-4		3-4	
Time	State0	State1	State0	State1	State0	State1	State0	State1
0	0	0	0	0	0	0	1	1
1	0.4	0.1	0.4	0.1	0.2	0.1	0	0
2	0.4	0.1	0.1	0.1	0.2	0.2	0	0
3	0	0.2	0.1	0.1	0.2	0.2	0	0
4	0.1	0.3	0.1	0.3	0.2	0.2	0	0
5	0.1	0.3	0.3	0.4	0.2	0.3	0	0

Fig. 1. An illustrative example

Two paths are available at node 1, namely path 1 (1-2-4) and path 2 (1-2-3-4). The distribution functions of the travel time on these paths are calculated using equation (3) and plotted in Fig. 2a. As shown, path 1 dominates path 2 in the first order at either of the SL states, because the distribution function of travel time on path 1 always lies above that on path 2. Thus, according to Definition 4, path 2 is dominated and thus not FSD-admissible.



a) Path 1 is dominated by path 2



b) Path 3 and 4 are FSD-admissible

Fig. 2. Distribution functions of path travel times

Fig. 2b plots the distribution functions of travel times on the two paths starting at node 2: path 3 (2-4), path 4 (2-3-4). As shown, path 4 dominates path 3 when the states on link 2-3 and 2-4 are 0 and 1 respectively (i.e., the first SL state in W_2). For the second SL state, however, path 3 dominates path 4. As a result, both paths are FSD-admissible since no path is dominated at all SL states.

3. FSD-based Dynamic Programming Formulation

Definition 4 suggests that a non-FSD-admissible path cannot be optimal at any state x, because for that state another path will always have an equal or lower travel time for any reliability by definition. It is worth noting, however, that an FSD-admissible path does not always guarantee optimality. Namely, an FSD-admissible path may never be shortest for all possible states and levels of reliability. To clarify this point, let us define FSD-optimality as follows.

Definition 5. A path k^{rs} is FSD-optimal if 1) it is FSD-admissible and 2) $\exists$ an $w \in W_r$ and one open interval $\Lambda \subset [0,T]$ with nonzero Lebesgue measure such that $u_{kx}^{rs}(b) \geq u_{ly}^{rs}(b), \forall b \in \Lambda, \forall l \neq k$, where $x = k^{rs}(w), \quad y = l^{rs}(w)$.

It follows from Definitions 4 and 5 that an FSD-optimal path is always FSD-admissible but not vice versa [2]. Let Γ_1^{is} and Ω_1^{is} be the sets of FSD-admissible and FSD-optimal paths between node i and s, respectively, and we have $\Omega_1^{is} \subseteq \Gamma_1^{is}$. Define, for every $w \in W_i$, a new function $U_w^{is}(\cdot)$ such that $\forall b \in [0,T]$, $U_w^{is}(b) = \max\{u_{kx}^{is}(b), \quad \forall k \in \Gamma_1^{is}, x = k^{is}(w)\}$. $U_w^{is}(\cdot)$ is called a *Pareto frontier function* at w, which represents optimal solutions of the reliable *a priori* shortest path problem for a given realization of SL states w. Clearly, once U_w^{is} is known, one can identify a path $\hat{k}^{rs} \in \Omega_1^{rs}$ such that $u_{\hat{k}x}^{is}(b) = U_w^{is}(b)$ for a given b and $x = \hat{k}^{is}(w)$.

Fig. 3 illustrates the concept using a small network where three paths are available at node 1, and W_1 includes two feasible SL states $w1$ and $w2$. The right hand side of the figure shows the distribution function of the three paths at $w1$ and $w2$. Because the first links of paths 2 and 3 are always at state x at all feasible SL states, we ignore the distribution functions for them at state y. As shown in the figure, all three paths are FSD-admissible. However, path 2 is not FSD-optimal at the SL state $w1$ because it does not contribute to the Pareto frontier. On the contrary, at the SL state $w2$, all three paths are FSD-optimal. Moreover, for a given time budget $b0$ as shown in the figure, the best path is 1 at the SL state $w1$ and 2 at $w2$.

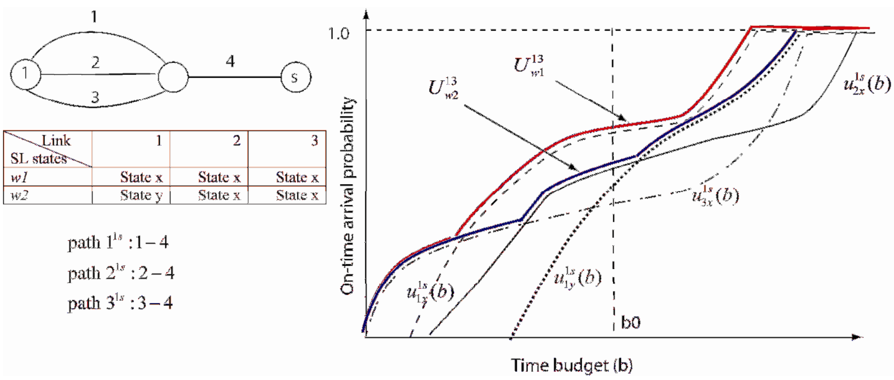


Fig. 3. An illustration of pareto-optimal functions

We are now ready to discuss how Definitions 4 and 5 may be used to facilitate the formulation and solution of the RASP problem. Nie and Wu (2008) proved that Bellman's principle of optimality applies to FSD-admissible paths when spa-

[2] FSD-optimality defined above is similar to the concept of *extreme non-dominance* in the literature (Henig 1985).

tial and temporal correlations are not considered. The following proposition extends that result to the correlated case.

Proposition 1. Subpaths of any FSD-admissible path must also be FSD admissible if the transition mapping ρ_{ij} is regulated for all link ij.

Proof: suppose that path k^{is} is FSD-admissible but one of its subpath k^{js} is not (note that $k^{is} = ij \Diamond k^{js}$). According to Definition 4, this means a path l^{js} exists such that $l^{js} \succ_1^w k^{js}$ for any $w \in W_j$. Let the first links of path l^{js} and k^{js} be jj' and jj'', respectively. Extending both paths l and k by link ij for link state x, equation (3) implies that $\forall b$

$$\begin{aligned} u_{lx}^{is}(b) - u_{kx}^{is}(b) &= \sum_y \rho_{ijj'}^{xy} \int_0^b u_{ly}^{js}(b-w)\, f_{ijx}^{T-b}(w)\,dw - \sum_z \rho_{ijj''}^{xz} \int_0^b u_{kz}^{js}(b-w)\, f_{ijx}^{T-b}(w)\,dw \\ &= \sum_y \rho_{ijj'}^{xy} \int_0^b \left(u_{ly}^{js}(b-w) - u_{ky}^{js}(b-w) \right) f_{ijx}^{T-b}(w)\,dw \geq 0 \end{aligned} ,(5)$$

The second equality holds because all ρ_{ij} are regulated, meaning that the transition probability from link ij to any successor link is identical (cf. Condition equation (2) in Definition 1). The inequality holds because $u_{ly}^{js}(b) \geq u_{ky}^{js}(b), \quad \forall b$ according to Definition 4. The above inequality must hold strictly for at least one open interval with nonzero Lebesgue measure because $l^{js} \succ_1^w k^{js}$ implies that $\exists$ such an interval for which $u_{ly}^{js}(b) > u_{ky}^{js}(b), \forall b$ in the interval. Therefore, $l^{is} \succ_1^w k^{is}$ for all w such that $x = l^{is}(w) = k^{is}(w)$. The second equality holds because both paths share the same first link ij .This in turn implies that k^{is} is not FSD-admissible, a contradiction.

Proposition 1 insures that RASP can be formulated as a general dynamic programming (GDP) problem (Carraway et al. 1990). Such a formulation aims to determine all FSD-admissible paths at any node i. It outperforms the aforementioned enumeration-based method because it excludes dominated paths from further consideration when constructing paths. A GDP formulation of the RASP problems reads.

Find $\Gamma_1^{is}, \forall i$ such that

$$\Gamma_1^{is} = \gamma_{\succ}^1 (k^{is} = k^{js} \Diamond ij \mid k^{js} \in \Gamma_1^{js}, \forall ij \in A), \forall i \neq s , \quad (6)$$

$$\Gamma_1^{ss} = 0^{ss} , \quad (7)$$

where $k^{js} \Diamond ij$ extends path k^{js} along link ij ; $\gamma_{\succ}^1(K)$ represents the operation which retrieves FSD-admissible paths from a set K using Definition 4; 0^{ss} is a dummy path representing the boundary condition.

Unlike in the conventional dynamic programming, solving equation (6) yields a set of FSD-admissible paths Γ_1^{is} as the optimal solution. Pareto frontier functions as well as the corresponding FSD-optimal path sets Ω_1^{is} can be constructed from

FSD-admissible paths, either concurrently or in a post-process [3]. Then, for a desired probability of on-time arrival and a realization of SL state, the shortest path can be read from the corresponding Pareto frontier function. In this case, the generated FSD-optimal paths and associated Pareto frontiers are used as a look-up table. An implicit assumption in the above formulation is that the desired arrival time at the destination s equals the longest inspected time budget T. Consequently, the problem has to be re-solved to update the look-up table whenever the destination and/or the desired arrival time are changed.

Despite it promises improved efficiency, the equations (6) and (7) do not insure a polynomial bound on the number of FSD-admissible paths. Indeed, the size of FSD-admissible path sets may grow exponentially with the network size. As a result, no algorithm can guarantee to generate all optimal paths in a polynomial time. We shall discuss this issue in greater details in the following section. For now, let's examine under what conditions these path sets are at least finite, namely, they do not contain cycles. For that purpose, we need first impose further restriction on our time-dependent probability density function f_{ijx}^{t}.

Definition 6. (*Stochastic FIFO)* A probability density function f_{ijx}^{t} is first-in-first-out (FIFO) consistent if its cumulative distribution function $F_{ijx}^{t}(\cdot)$ satisfies the following condition

$$t_1 \leq t_2 \Rightarrow F_{ijx}^{t_1}(b) \geq F_{ijx}^{t_2}(b), \forall b \in [0,T]\,, \tag{8}$$

The FIFO-consistency condition ensures that those who depart earlier have a higher probability of arriving earlier. This is a stochastic extension of the FIFO principle often discussed in the literature of dynamic traffic assignment models (Astarita 1996). The following lemma is a direct result of FIFO consistency.

Lemma 1. Any path travel time distribution function $u_{kx}^{is}(b)$ is non-decreasing if f_{ijx}^{t} is FIFO consistent for any $ij \in A, x \in \chi$.

Proof: For any node i that directly connects to the destination s, the monotonicity directly follows from Definition 6: $u_{kx}^{is}(b_1) \geq u_{kx}^{is}(b_2),\ \forall b_1 \geq b_2$ because the departure time $T - b_1$ is smaller (earlier) than the departure time $T - b_2$.

If node i does not directly connect with s, then without loss the generality let's consider a path $i \rightarrow j \rightarrow s$ where node j directly connects to s. According to equation (3), given $\forall b_1 \geq b_2$, we have:

[3] Which way is adopted depends on the context. For individual routing inquiry, postprocess may be preferred since most Pareto-frontier functions will not be used.

$$
\begin{aligned}
& u_{kx}^{is}(b_1) - u_{kx}^{is}(b_2) \\
&= \sum_y \rho_{ijl}^{xy} \left(\int_0^{b_1} u_{ky}^{js}(b_1 - w) f_{ijx}^{T-b_1}(w)dw - \int_0^{b_2} u_{ky}^{js}(b_2 - w) f_{ijx}^{T-b_2}(w)dw \right) \\
&= \sum_y \rho_{ijl}^{xy} \left(u_{ky}^{js}(b_1 - w) F_{ijx}^{T-b_1}(w) |_0^{b_1} + \int_0^{b_1} u_{ky}^{js}{}'(b_1 - w) F_{ijx}^{T-b_1}(w)dw - u_{ky}^{js}(b_2 - w) F_{ijx}^{T-b_2}(w) |_0^{b_2} - \int_0^{b_2} u_{ky}^{js}{}'(b_2 - w) F_{ijx}^{T-b_2}(w)dw \right), \quad (9) \\
&= \sum_y \rho_{ijl}^{xy} \left(\int_0^{b_1} f_{jsy}^{T-b_1+w}(b_1 - w) F_{ijx}^{T-b_1}(w)dw - \int_0^{b_2} f_{jsy}^{T-b_2+w}(b_2 - w) F_{ijx}^{T-b_2}(w)dw \right) \\
&\geq \sum_y \rho_{ijl}^{xy} \left(\int_0^{b_2} f_{jsy}^{T-b_1+w}(b_1 - w) F_{ijx}^{T-b_2}(w)dw - \int_0^{b_2} f_{jsy}^{T-b_2+w}(b_2 - w) F_{ijx}^{T-b_2}(w)dw + \int_{b_2}^{b_1} f_{jsy}^{T-b_1+w}(b_1 - w) F_{ijx}^{T-b_1}(w)dw \right) \\
&= \sum_y \rho_{ijl}^{xy} \left(\int_0^{b_2} [f_{jsy}^{T-b_1+w}(b_1 - w) - f_{jsy}^{T-b_2+w}(b_2 - w)] F_{ijx}^{T-b_2}(w)dw + \int_{b_2}^{b_1} f_{jsy}^{T-b_1+w}(b_1 - w) F_{ijx}^{T-b_1}(w)dw \right) \\
&\geq \sum_y \rho_{ijl}^{xy} \left(F_{ijx}^{T-b_2}(0) \int_0^{b_2} [f_{jsy}^{T-b_1+w}(b_1 - w) - f_{jsy}^{T-b_2+w}(b_2 - w)]dw + \int_{b_2}^{b_1} f_{jsy}^{T-b_1+w}(b_1 - w) F_{ijx}^{T-b_1}(w)dw \right) \\
&\geq \sum_y \rho_{ijl}^{xy} \left(\int_{b_2}^{b_1} f_{jsy}^{T-b_1+w}(b_1 - w) F_{ijx}^{T-b_1}(w)dw \right) \geq 0
\end{aligned}
$$

Note that given $\forall b_1 \geq b_2 \geq w \geq 0$, we have $F_{ijx}^{T-b_1}(w) \geq F_{ijx}^{T-b_2}(w) \geq F_{ijx}^{T-b_2}(0)$ and $\int_0^{b_1} f_{jsy}^{T-b_1+w}(b_1 - w)dw \geq \int_0^{b_2} f_{jsy}^{T-b_2+w}(b_2 - w)dw) \geq 0$ because of the condition of FIFO and because the CDF of the link traversal time is always monotone.

The reader is referred to Nie and Wu (2008) for a numerical example which demonstrates the existence of non-monotone path traversal time distribution functions when the FIFO-consistency is not enforced. Another main result now follows.

Proposition 2. An FSD-admissible must not contain any cycle if all f_{ijx}^t are FIFO consistent.

Proof: we need to show that a path with cycles is always dominated by the same path without the cycles, and thereby is not FSD-admissible. We only consider the case illustrated in the Fig. 4, where a single cycle $i \rightarrow i' \rightarrow i$ is added to path l^{is} to form a new path k^{is}. More complicated cases can be reduced to this simple one. According to the principle of the optimality, if the subpath k^{is} is dominated, any of its extensions is too. Thus, the argument can be applied to the situation where the cycle occurs in the middle of path. Multiple cycles can be treated once a time and thus reduced to the case described in Fig. 4.

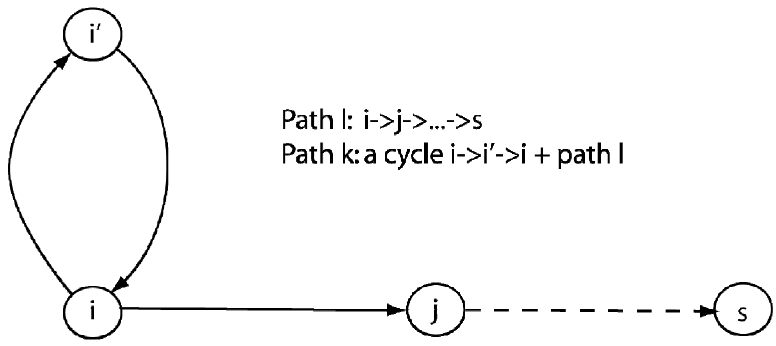


Fig. 4. An illustration of cyclic paths

Consider an SL state w where the states on link ii' and link ij are x and y, respectively. Without loss of generality, let us assume that x and y are only possible link states, i.e., $\chi = \{x, y\}$. From equation (3), for the path with the cycle we have

$$u_{kx}^{is}(b) = \rho_{ii'i}^{xx} \int_0^b u_{kx}^{i's}(b-w) f_{ii'x}^{T-b}(w)dw + \rho_{ii'i}^{xy} \int_0^b u_{ky}^{i's}(b-w) f_{ii'x}^{T-b}(w)dw, \quad (10)$$

$$\leq \left[\rho_{ii'i}^{xx} u_{kx}^{i's}(b) + \rho_{ii'i}^{xy} u_{ky}^{i's}(b) \right] \int_0^b f_{ii'x}^{T-b}(w)dw. \quad (11)$$

The inequality follows from Lemma 1. When the state on link $i'i$ is x,

$$u_{kx}^{i's}(b) = \rho_{i'ij}^{xx} \int_0^b u_{lx}^{is}(b-w) f_{i'ix}^{T-b}(w)dw + \rho_{i'ij}^{xy} \int_0^b u_{ly}^{is}(b-w) f_{i'ix}^{T-b}(w)dw. \quad (12)$$

For the SL state w, the state on link ij is fixed as y. Thus, the above relationship is simplified as

$$u_{kx}^{i's}(b) = \int_0^b u_{ly}^{is}(b-w) f_{i'ix}^{T-b}(w)dw \leq u_{ly}^{is}(b) \int_0^b f_{i'ix}^{T-b}(w)dw. \quad (13)$$

Again, the inequality holds because of Lemma 1. Similarly, we have

$$u_{ky}^{i's}(b) \leq u_{ly}^{is}(b) \int_0^b f_{i'iy}^{T-b}(w)dw. \quad (14)$$

Replacing $u_{kx}^{i's}$ and $u_{ky}^{i's}$ in (11) with (13) and (14) yields

$$\begin{aligned} u_{kx}^{is}(b) &\leq \left[\rho_{ii'i}^{xx} u_{ly}^{is}(b) \int_0^b f_{i'ix}^{T-b}(w)dw + \rho_{ii'i}^{xy} u_{ly}^{is}(b) \int_0^b f_{i'iy}^{T-b}(w)dw \right] \int_0^b f_{ii'x}^{T-b}(w)dw \\ &= u_{ly}^{is}(b) \left[\rho_{ii'i}^{xx} \int_0^b f_{i'ix}^{T-b}(w)dw + \rho_{ii'i}^{xy} \int_0^b f_{i'iy}^{T-b}(w)dw \right] \int_0^b f_{ii'x}^{T-b}(w)dw \\ &\leq u_{lx}^{is}(b) \left[\rho_{ii'i}^{xx} + \rho_{ii'i}^{xy} \right] \int_0^b f_{ii'x}^{T-b}(w)dw = u_{ly}^{is}(b) \int_0^b f_{ii'x}^{T-b}(w)dw \leq u_{ly}^{is}(b) \end{aligned} \quad .(15)$$

Note that the first inequality follows from Lemma 1, the second equality holds because of the first regulatory condition equation (1), and finally the second and third inequalities hold because $\int_0^T f_{ijx}^t(w)dw \leq 1, \forall ij, x$. Finally, there are always Λ such that $\int_0^b f_{ijx}^t(w)dw < 1$ if $b \in \Lambda$. From Definition 3 this implies path $l^{is} \succ_1^w k^{is}$. One can show that this holds for all possible SL states w. Consequently, k^{is} is not FSD-admissible. Therefore, any cyclic path cannot be FSD-admissible for FIFO-consistent probability density functions.

The acyclicity of FSD-admissible path not only insures the finiteness of the problem, but also allows one to exclude any cyclic path from further consideration. This has some useful implication in terms of algorithmic design, as we shall discuss in the next section.

4. Solution Algorithm and Complexity

The solution algorithm to the RASP problem consists of two main building blocks: the evaluation of the convolution integral in the recursive equation (3), and the iterative construction of FSD-admissible path sets through dynamic-programming techniques (such as labeling operations). Evaluating convolution integrals usually calls for numerical procedures because the path travel time distribution function does not typically have a closed form. Consequently, the problem should be discretized so that the convolution can be carried out efficiently. Such a discrete scheme has been studied in Nie and Wu (2008), and can be slightly revised to address the RASP problem with correlations. A brief description of the scheme is given below for the sake of completeness.

Let $T = L\phi$, where ϕ is the length of unit time budget (which also equals, in our setting, to the length of departure time interval). The time budget b is treated in a discrete manner, i.e., $b = 0, \phi, 2\phi, \cdots, L\phi$. For any link ij, the probability mass function (PMF) $P^t_{ijx}(b)$ of c_{ij} on state x may be obtained from its probability density function $f^t_{ijx}(b)$ (provided such a function is available in a continuous form) as follows:

$$P^t_{ijx}(b) = \begin{cases} \int_b^{b+\phi} f^t_{ijx}(w)dw & b = 0, \phi, ..., (L-1)\phi \\ \int_b^{\infty} f^t_{ijx}(w)dw & b = L\phi \\ 0 & \text{otherwise} \end{cases} . \tag{16}$$

Accordingly, equation (3) is discretized as:

$$u^{is}_{kx}(b) = \sum_y \rho^{xy}_{ij\ell} \sum_{\eta=0}^{b} u^{js}_{ky}(b - \eta\phi) P^{T-b}_{ijx}(\eta\phi), \forall b = 0, \phi, ..., L\phi, \forall i \neq s, \forall x \in \chi, \tag{17}$$

$$\rho^{xy}_{iss} = 1, \forall x, y \in \chi, u^{ss}_{k0}(b) = 1, \forall b = 0, \phi, ..., L\phi, \tag{18}$$

We are now ready to present a label-correcting algorithm that constructs the FSD-admissible path set according to equation (5). The framework of the algorithm is in general similar to those discussed in Miller-Hooks (1997), Miller-Hooks and Mahmassani (2000) and Nie and Wu (2008). The major difference lies in how the operator $\gamma^1_{\succ}$ in equation (5) is implemented. For the ease of reference, the algorithm is tagged as C-FSD-LC where C and LC stand for "Correlated" and "Label-Correcting" respectively. A description of the algorithm now follows.

Algorithm C-FSD-LC

Step 0: Initialization. set pareto frontier function $U^{is}_w(b) = 0, \forall b \in [0,T], w \in W_i, i \in N/\{s\}$. Set $u^{ss}_w(b) = 1$ $\forall b \in [0,T], w \in W_s$. Note that by definition W_s contains only one dummy state 0, i.e., $W_s = \{0\}$. Initialize path 1^{ss} at the destination s such that $u_{10}^{ss}(b) = 1, \forall b$, Set $\Gamma_1^{ss} = \{1^{ss}\}$, and $Q = \{1^{ss}\}$

Step 1: Check optimality. If $Q = \varnothing$, terminate the procedure, the optimal solution is found; otherwise, go to Step 2.

Step 2: Take the first path k^{js} from Q, set $Q = Q / k^{js}$. For each link $ij \in A$, repeat Steps 2.1 - 2.3. Then go back to Step 1.

Step 2.1: Call Procedure CYCLE-CHECK. If node i is not on path k^{js}, go to step 2.2.

Step 2.2: Set $l = |\Gamma_1^{is}| + 1$, create a new path l^{is}, calculate u_{lx}^{is} for all $x \in \chi$.

Step 2.3: If $l = 1$, set $\Gamma_1^{is} = \{l^{is}\}, U_w^{is}(b) = u_{lx}^{is}(b), \forall b$ and for any w such that $x = l^{rs}(w)$ and $\omega(l^{is}) = k^{js}$, where $\omega(\cdot)$ is a subpath operator used to track paths. Otherwise, call Procedure FSD-CHECK. If l^{is} is FSD-admissible, set $\omega(l^{is}) = k^{js}$ and update $Q = Q \cup l^{is}$.

We note that the above procedure is similar to the algorithm presented in Nie and Wu (2008), except that in Step 2.2 the path distribution function has to be evaluated for every $x \in \chi$ whenever a new path is generated. CYCLE-CHECK in Step 2.1 excludes any cyclic paths from further consideration based on Proposition 2. The check can be performed by recursively applying the subpath operator $\omega(\cdot)$ until the head of the subpath reaches the destination node s, as shown in the following.

Procedure CYCLE-CHECK

Inputs: path l^{js} and node i such that $ij \in A$

Return: a boolean value CR indicating whether or not i has been traversed by l^{js}.

Step 0: Set $k^{ps} = \omega(l^{js})$, if $p = i$, CR = TRUE, stop; else if $p = s$, CR = FALSE, stop; otherwise goto Step 1.

Step 1: Set $l^{js} = k^{ps}$, go to Step 0.

In the worst case, the above operation has to access each node in the network no more than once. We note that CYCLE-CHECK is not a necessary operation since a cyclic path is not FSD-admissible and hence will fail to pass FSD-CHECK eventually. However, preventing such paths from entering FSD-CHECK may be preferable from a computational point of view. We now show how to determine whether or not a new path is FSD-admissible and update the Pareto frontier functions simultaneously.

Procedure FSD-CHECK

Inputs: a new path l^{is} and the corresponding $u_{lx}^{is}(b), \forall x, b = 0, \phi, \ldots L\phi$, a set of FSD-admissible paths Γ_1^{is}, the set of SL states W_i, and Pareto frontier functions $U_w^{is}, \forall w$.

Return: a Boolean value **FSD** indicating whether or not l^{is} is FSD-admissible, updated U_w^{is} and Γ_1^{is}.

Step 0: Initialization. Set FSD = FALSE.

Step 1: Update all pareto frontier functions. For every $w \in W_i$, do the following. Compare $U_w^{is}(b)$ with $u_{lx}^{is}(b), x = l^{is}(w), \forall b = 0, \phi, \ldots L\phi$. Whenever $U_w^{is}(b) < u_{lx}^{is}(b)$, set $U_w^{is}(b) = u_{lx}^{is}(b)$ and FSD = TRUE. If $U_w^{is}(b) < u_{lx}^{is}(b)$ for all b and $x = l^{is}(w)$ for all w, then let $\Gamma_1^{is} = \varnothing$, and go to Step 3; otherwise go to Step 2.

Step 2: For every path $k^{is} \in \Gamma_1^{is}$, repeat Steps 2.1 to 2.2.

Step 2.1: Set $\iota_l = 0, \iota_k = 0$. For every $w \in W_i$, compare $u_{lx}^{is}(b)$ and $u_{ky}^{is}(b)$. If $l^{is} \succ_1^w k^{is}$ (i.e., $u_{lx}^{is}(b) \ge u_{ky}^{is}(b), \forall b$ and $u_{lx}^{is}(b) > u_{ky}^{is}(b)$ for some b, where $x = l^{is}(w), y = k^{is}(w)$, set $\iota_l = \iota_l + 1$, check the next w; else if $k^{is} \succ_1^w l^{is}$, set $\iota_k = \iota_k + 1$, check the next w; otherwise, go to Step 2.2.

Step 2.2: If $\iota_l = |W_i|$, set $\Gamma_1^{is} = \Gamma_1^{is} / k^{rs}$, check next path; else if $\iota_k = |W_i|$, set FSD = FALSE and then go to Step 3; otherwise, check the next path in Γ_1^{is}.

Step 3: If FSD = TRUE, set $\Gamma_1^{is} = \Gamma_1^{is} \cup l^{is}$, stop.

Note that any path contributing to the frontier cannot be dominated. The following result establishes the correctness and finiteness of the label-correcting algorithm.

Theorem 1. Algorithm C-FSD-LC terminates after a finite number of steps and yields a set of FSD- admissible paths at each node i.

Proof: The algorithm ensures that no path can enter the candidate list more than once and Proposition 2 guarantees all cyclic paths will be dominated by their acyclic counterparts. The finiteness of the algorithm then follows from the fact that a network has a finite number of paths. Through FSD-CHECK, only acyclic and FSD-admissible paths will be kept at termination. Therefore the retained paths must form Γ_1^{is}.

The complexity of Algorithm C-FSD-LC depends on the size of the FSD-admissible paths, which is in turn bounded only by the number of paths. Since the number of paths in a network grows exponentially with the number of nodes n (roughly n^{n-1} in the worse case), no polynomial algorithm exists for the RASP problem. Consequently, large-scale instances of RASP may be intractable. The intractability of multi-criteria shortest path problems (of which RASP is a variant) is well known (e.g. Hansen 1979; Henig 1985; Miller-Hooks and Mahmassani 2000). In the following, we provide an analysis of the complexity of the algorithm and show how it can be improved using various approximations/simplifications.

Proposition 3. Algorithm C-FSD-LC runs in a non-deterministic polynomial time $O(|\chi|^2 Lmn^{2n-1} + mn^n |\chi|^{n^2} L^2)$, where L is the number of discrete time points, m is number of links, n is number of nodes, and $|\chi|$ is the number of possible link states.

Proof: First of all, if each link can take $|\chi|$ different values, the maximum possible size of the SL states at each node, denoted as W, is bounded by $|\chi|^m$, or more precisely $|\chi|^{n^2}$. In the worse case, the algorithm may have to examine all possible paths for any origin-destination pair is. There are $n \times n^{n-1} = n^n$ paths in total. For each path, every extension may involve up to m links. Therefore Step 2 of the algorithm may be executed mn^n time. Within Step 2, for each path $O(n)$ operations are required to check the acyclicity and $O(WL^2)$ operations are required to calculate convolution integral. Thus, the work involved in Steps 2.1-2.2 is bounded by $O(mn^n(n + WL^2))$.

In Procedure FSD-CHECK, $O(WL)$ operation is consumed in Step 1. In the worst case, all paths in Γ_1^{is} (bounded by n^{n-1}) needs to be compared with the new path for all possible SL states. While each comparison should be conducted once for each SL state w, we note that the number of possible link state combinations for any two paths involved in the comparison should be bounded by $|\chi|^2$ instead of W. Thus $|\chi|^2 Ln^{n-1}$ operations are needed. Finally, the complexity of the algorithm is in the order of $O(mn^n[n + |\chi|^{n^2} L^2 + |\chi|^{n^2} L + |\chi|^2 Ln^{n-1}])$ $\approx O(|\chi|^2 Lmn^{2n-1} + mn^n |\chi|^{n^2} L^2)$.

In reality, the maximum number of possible SL states W should be much smaller than the formidable bound $|\chi|^{n^2}$, given the likely correlations between links that share the same starting node. For example, if one assumes that all links starting from the same node have the same state, W can be bounded by $|\chi|$. The complexity of the algorithm can thus be reduced to $O(|\chi| Lmn^{2n-1})$ (assuming that $L \ll n^n$). Also note that n^{n-1} is a rather conservative bound for the number of paths $|K^{rs}|$. In sparse networks often seen in surface transportation systems, it is expected that $|K^{is}| \ll n^{n-1}$. Nevertheless, whether or not the number of FSD-admissible paths has a polynomial bound is an unresolved issue. Henig (1985) showed that the expected number of FSD-admissible paths $|\Gamma_1^{is}|$ can be bounded by $\log(|K^{is}|)$ when $L = 2$ and $W = 1$. Nie and Wu (2008) postulated, based on numerical experiments, that $|\Gamma_1^{is}|$ is bounded by $L\log(|K^{is}|)$ when $L > 2$ and $W = 1$ thanks to the monotonicity of distribution functions. However, how fast

$|\Gamma_1^{is}|$ grows with W becomes a more difficult question to answer even through numerical experiments, let alone using theoretical justification. We defer the investigation of this problem to a further study.

A possible approximation that leads to a better complexity is based on the assumption that the principle of optimality is applicable to FSD-optimal paths. If only FSD-optimal paths are to be retained, the number of paths is bounded by WL. Moreover, the time-consuming pair wise comparison in Step 3 of Procedure FSD-CHECK can be skipped because all paths that do not contribute to the frontier is automatically dominated under this assumption. As a result, the complexity of the algorithm is reduced to $O(mWL[n+WL^2+WL]) \approx O(mW^2L^3)$. This remains to be non-deterministic polynomial should W be bounded by $|\chi|^{n^2}$. However, if the aforementioned restriction on W (i.e., all outgoing links should have the same state) is imposed, a polynomial bound $O(m|\chi|^2 L^3)$ can be achieved. When $|\chi|=1$, the bound degrades to the uncorrelated case $O(mL^3)$. As a final note, the approximation based on FSD-optimality does not necessary yield correct Pareto frontier functions, because subpaths of a Pareto-optimal path may not be Pareto-optimal. However, our experiments suggest that the approximation produce results that closely resemble actual Pareto frontiers in most cases.

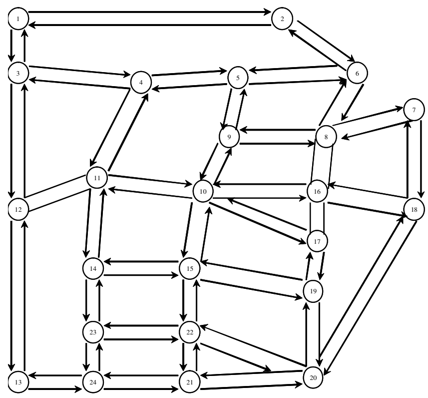
5. Numerical Examples

The topology of two "typical" road networks, namely the Sioux Falls (SF) network and the Chicago Sketch (CK) network, is used to construct test routing problems (see Fig. 5). Our experiments only use their topology. The algorithm C-FSD-LC was coded using MS-C++ and tested on a Windows-XP (64) workstation with two 3.00 GHz Xeon CPUs and 4G RAM.

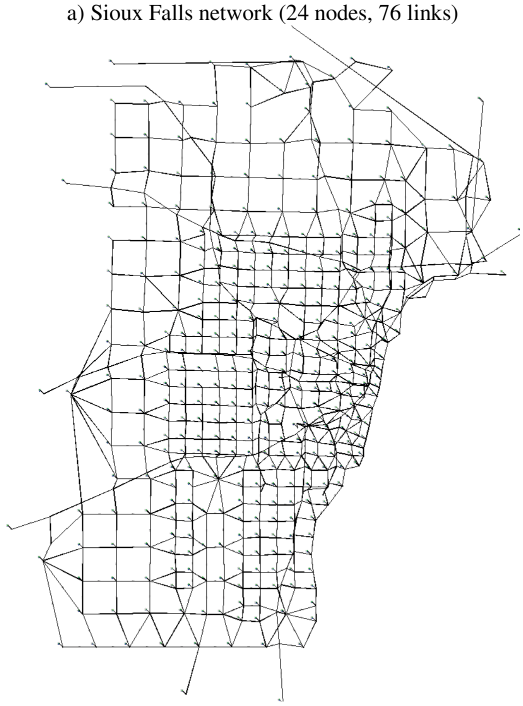
The link travel time c is assumed to follow a gamma distribution:

$$f_{ij}^t(c) = \frac{1}{\alpha^\beta \Gamma(\beta)}(c-c_L)^{\beta-1} e^{-(c-c_L)/\beta}, \tag{19}$$

where α, β and c_L are parameters, and $\Gamma(\cdot)$ is the gamma function. For each link ij, the probability mass function $P_{ijx}^t(\cdot)$ is generated using equation (15). In the distribution equation (18), the minimum possible link traversal time is given by c_L. We choose the maximum possible time c_U such that $\int_{c_U}^{\infty} f_{ij}^t(w)dw < \varepsilon$ .Thus, for a given ε, we can select the α, β, c_L such that possible travel times range between (c_L, c_U). For the sake of convenience, we shall use parameters c_L and c_U to control link PMFs in our experiments. Moreover, by altering the values of ε we can generate different PMFs from the same c_L and c_U.



a) Sioux Falls network (24 nodes, 76 links)



b) Chicago sketch (933 nodes, 2950 links)

Fig.5. Test networks

For simplicity, in all numerical examples links sharing the starting node are required to be at same states. Namely, the size of SL states equals the size of possible link states ($|W_i|=|\chi|$). Unless otherwise specified, the transition probabilities are randomly drawn from a proper uniform distribution such that the regulatory conditions (1) and (2) are met. In this experiment, transition probability matrices on links with the same ending node are not forced to be identical for convenience. As noted in Section 2, whether or not this restriction is imposed should not affect the applicability and performance of the algorithm. Throughout the example, the analysis period is assumed to be 100 (that is to say, the total time budget or the desired arrival time $T = 100$). The analysis period is divided into 100 discrete intervals, i.e., $\phi = 1$.

We first provide an illustrative example that considers time-varying probability mass functions, and then conduct two computational studies focused on the performance of the proposed algorithms.

5.1. An Example with Time-dependent Probability Mass Functions

Let $\chi = \{0,1\}$, where 0 and 1 represent uncongested and congested traffic conditions respectively. In our setting, the earliest departure time is 0 (when the time budget is 100), and the latest departure time is 100 (when the time budget is 0). Probability mass functions are allowed to vary from one departure time period to another. For the illustrative purpose, only three such periods are defined, namely, $T^1 = [0,30], T^2 = [31,60], T^3 = [61,100]$. Clearly, it is straightforward to consider more refined time resolutions as far as data availability justifies it. The range of possible realization of link traversal times is then defined as below.

$$\text{uncongested}: (c_L, c_U) = \begin{cases} (0,5) & t \in T^1 \\ (6,10) & t \in T^2 \\ (11,15) & t \in T^3 \end{cases}, \text{congested}: [c_L, c_U] = \begin{cases} (8,20) & t \in T^1 \\ (16,25) & t \in T^2 \\ (20,30) & t \in T^3 \end{cases}. \quad (20)$$

We emphasize that the probability mass functions in this experiment are carefully set to satisfy the stochastic FIFO condition (see Definition 6).

For OD pair $(5,24)$, Algorithm C-FSD-LC identified three FSD-admissible paths: Path 3: $5 \to 4 \to 11 \to 14 \to 23 \to 24$; Path 6: $5 \to 4 \to 11 \to 12 \to 13 \to 24$; and Path 11: $5 \to 4 \to 3 \to 12 \to 13 \to 24$. Fig. 6 shows the profiles of three paths at both uncongested and congested SL states. We found that Path 3 is the most reliable path in either SL state, so long as the departure time is earlier than 30. Path 6 outperforms path 3 when the departure time is later than 43 (uncongested) or 30 (congested). Although Path 11 is included as an FSD-admissible path, it is not FSD-optimal. Fig. 6 demonstrates a jump on the distribution functions around the departure time $t = 30$, which obviously coincides with the boundary between T^1 and T^2.

5.2. Computational Performance of Algorithm C-FSD-LC

Recall that links with the same starting node are assumed to have the same state for any SL state in our experiments. Thus, the complexity of Algorithm C-FSD-

LC is $O(|\chi|^2 Lmn^{2n-1})$ as shown in Section 4. This section demonstrates how the algorithm actually performs for different networks and number of link states. All link traversal time distributions are assumed to be time-independent because introducing time-dimension is irrelevant to computational performance.

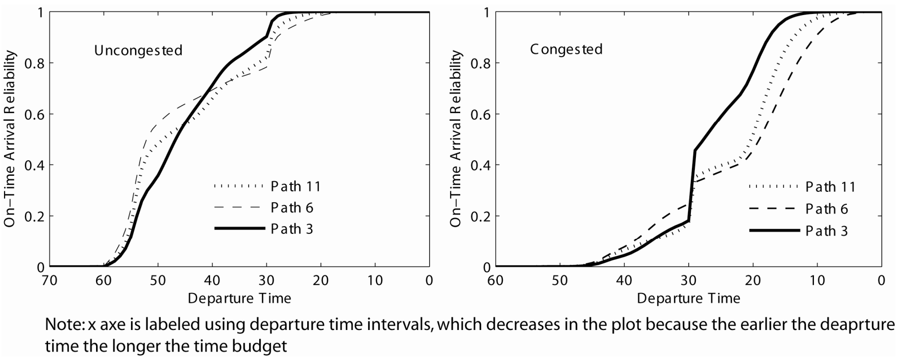


Fig.6. Traversal time distributions of FSD-admissible paths (SF Network, OD pair (5, 24))

Let $\{0,1,2,3\}$ represent four levels of congestion, where "0" refers to the least congested level, and "3" refers to the most congested level. In Scenarios I, II, III and IV, the set of all possible link states χ is set as $\{0\},\{0,1\}, \{0,1,2\},\{0,1,2,3\}$, respectively. Also, corresponding to each of the possible link states $x = 0,1,2,3$, the range of link traversal times $(c_L, c_U) = (0,10),(11,15),(16,20),(21,25)$, respectively.

In our test, 10 nodes are randomly selected from each network (SF and CK) as the destination for each of the four scenarios. Then FSD-admissible paths and Pareto frontier functions are solved using Algorithm C-FSD-LC for every destination. In each scenario, average CPU times (over all 10 runs), maximum and average numbers of FSD-admissible paths (for all nodes and runs) were generated as the measure of the computational performance of the algorithm. These results are reported in Table 1.

Table 1. Computational performance of algorithm C-FSD-LC in four Scenarios

Network		I ($\|\chi\|=1$)	II ($\|\chi\|=2$)	III ($\|\chi\|=3$)	IV ($\|\chi\|=4$)
SF	Time (sec)	0.05	1.78	2.65	5.30
(24 nodes)	MaxFSD	8	12	8	11
(76 links)	AveFSD	3.7	5.1	3.8	4.6
CK	Time (sec)	11.18	42.39	116.39	418.17
(933 nodes)	MaxFSD	53	60	90	120
(2950 links)	AveFSD	10.6	10.7	14.4	17.4

Note: MaxFSD - maximum number of FSD-admissible paths at any node; AveFSD - average number of FSD-admissible paths at all nodes

The average numbers of FSD-admissible paths (AveFSD) are reasonably small in almost all cases. The good news is that it does not seem to increase exponentially with the network size. The CK network is about 40 times larger than the SF network in the node size, but its AveFSD index is only 3 - 4 times higher. Neither did AveFSD seem to increase exponentially with $|\chi|$. In the Chicago Sketch network, when $|\chi|$ increases from 1 to 4, AveFSD barely doubles in both networks. In the Sioux Fall network, the increase of $|\chi|$ does not necessarily cause the increase of MaxFSD and AveFSD. Note that both MaxFSD and AveFSD drops when $|\chi|$ increases from 2 to 3. Apparently, the size of FSD-admissible paths depends on not only link states and network topology, but also other factors such as transition probability.

On the other hand, the CPU time consumed appears to be better agreeing the given formula of complexity. Approximately, the CPU time for the same network increases at a rate proportional to the product of the square of $|\chi|$ and the square of AveFSD. Also, it seems that the CPU time is more sensitive to $|\chi|$ in larger networks.

5.3. The Performance of an Approximation Algorithm

As mentioned at the end of Section 4, a polynomial complexity can be achieved if the applicability of Bellman's principle of optimality extends to the FSD-optimal paths. This assumption suggests the possibility of removing all paths which do not contribute to Pareto frontiers from further consideration. In turn, the comparison between the new path and the FSD-admissible paths that do not contribute to the frontier can be skipped. Although appealing from a computational point of view, we caution that the approximation algorithm may miss some FSD-optimal paths and thereby not yield the optimal Pareto frontiers. The main purpose of this experiment is thus to measure to what extent the results from the approximation algorithm deviate from the correct ones.

In this test, only the CK network is considered. Also, $\chi = \{0,1\}$ and link traversal time distributions are time-independent. We randomly select 20 nodes as destinations and obtain Pareto frontiers using both algorithms for each destination. The indexes of computational performance are averaged over all 20 runs, as reported in Table 2.

Define for each destination a Pareto gap $\Delta U_w^s = \max\{|U_w^{is}(b) - V_w^{is}(b)| : \forall b, i\}$, where U_w^{is} and V_w^{is} are Pareto frontiers solved by Algorithm C-FSD-LC and the approximation algorithm respectively. ΔU_w^s measures the error of the approximation algorithm in terms of maximum on-time arrival probability. Twenty Pareto gaps were generated for either of the two SL states. Table 2 gives the gaps at the 30th, 60th and 90th-percentiles.

Table 2. The comparison between C-FSD-LC and the approximation algorithm

Algorithm	CPU Time	Max # FSD path	Ave # FSD path		Gap of Pareto Frontiers 90th	60th	30th
C-FSD-LC	31.52	35	6.37	SL 0	5.310E-03	3.390E-04	0.000E+00
Approximation	18.80	21	4.79	SL 1	5.616E-03	1.954E-03	2.804E-05

Note: 5.310E-03 under 90th (percentile) at SL 0 means that U_0^s is no larger than 0.0053 in 18 out of the 20 runs (90%) when the SL state is 0.

The computational advantage of the approximation algorithm is clear and expected. It ran roughly two times faster than its counterpart, apparently because it only generated about two thirds of FSD-admissible paths. However, despite that the approximation algorithm missed almost one third of the FSD-admissible paths, it yielded Pareto frontiers similar to those from the exact algorithm. A reason may be that a large portion of the excluded FSD-admissible paths are not FSD-optimal. Note that U_0^s at the 90th-percentile are less than 1% for both states. That is to say, in 90% of all case, users who take the best path generated from the approximation algorithm would not worsen their on-time arrival probability by more than 1% even in the worse case. Such a discrepancy is negligible in many if not most practical applications.

6. Concluding Remarks

The advances in cyberinfrastructure and communication technology are making available real-time traffic data at an unprecedented scale, and hence promising to reveal to greater details random patterns of road travel times. The reliable *a priori* shortest path problem (RASP) studied in this paper makes use of such information to find most reliable paths in a stochastic and time-dependent network. Reliable routes help risk averse travelers maximize their likelihood of arriving on-time or minimize the buffer time reserved for uncertainty. The results from this paper may also lead to improved traffic assignment models that integrate supply uncertainty and drivers' routing behavior pertinent to reliability.

The overarching goal of the present paper is to introduce a correlation structure into the reliable *a priori* shortest path problem. For one thing, correlated road travel times better reflect the realism of highway transportation systems. Perhaps more important, a correlated model makes it possible to integrate real-time and historical data. Specifically, due to correlations, shortest paths are not only determined by historical travel times, but also conditional on real-time observations revealed up to the time when the routing decision has to be made. In reality these observations may not be restricted to traffic conditions. Other factors, such as weather or seasonal effects, may also affect the distribution of road travel times. While an ideal routing model ought to take all these factors into consideration, incorporating such a general correlation structure presents great challenges to both the modeling capability and data availability. To strike compromise between realism and tractability, this paper models spatial correlations among link traversal times by allowing random probability density functions to be conditional on link states. As demonstrated before, these correlations are spatially limited by the Markovian property of the link states.

A key result of the paper is the formulation of the RASP problem with the aforementioned correlation structure as a general dynamic programming (DP) problem. This formulation is built upon two theoretical developments that are extensions of those presented in Nie and Wu (2008). First, conditions are proposed to regulate the transition mapping of link states such that Bellman's principle of optimality can be applied to FSD-admissible paths, which are optimal solutions to the RASP problem. Second, cyclic paths are excluded from the set of FSD-admissible paths based on the assumption that all random link traversal times are FIFO consistent in the stochastic sense. To solve the DP formulation, a label correcting algorithm is designed, which is similar to those proposed in Miller-Hooks (1997) and Nie and Wu (2008). However, FSD-admissible paths are determined differently because these paths are now dependent on link states. The algorithm, like its counterparts in the aforementioned studies, is generally non-deterministic polynomial in its complexity. Approximation algorithms with polynomial complexity may be achieved when further assumptions are made to the correlation structure and to the applicability of the DP formulation.

According to preliminary experiments conducted in this study, the correlated RASP problem may be practically tractable. The number of FSD-admissible paths does not demonstrate an exponential growth with network size. This behavior is expected and has been demonstrated in previous studies (Miller-Hooks 1997; Nie and Wu 2008). However, the number of link states ($|\chi|$) also does not seem to exponentially affect the average size of FSD-admissible paths. This is a somewhat unexpected good news, because conceivably correlations will make it harder to establish dominance among paths, and more likely render non-deterministic polynomial algorithmic behavior. However, the computational overhead still increases super-linearly with $|\chi|$. As a result, a practically useful model may have to restrict $|\chi|$ to be a rather small number. The experiments also suggest that the approximation algorithm mentioned above provides an appealing alternative to the exact algorithm. It gains substantial computational efficiency as expected, while generating results practically comparable to those from the exact algorithm. We caution that, however, the above findings are not conclusive, and more extensive numerical studies are required to prove or disprove them.

There are other aspects in the future development of the RASP that call for considerable efforts. First, it is important to explore the tractability of the RASP when the restrictions on the correlation structure imposed in Section 2 are relaxed. Secondly and perhaps more interestingly, higher-order stochastic dominance relationship (Levy and Hanoch 1970; Levy 1992) can be used to redefine the optimality. The second-order stochastic dominance (SSD), for example, better captures an individual's risk-averse behavior because it maximizes the utility for an individual with any concave nondecreasing utility function. SSD is also appealing from a computational perspective because it typically yields a significantly smaller set of non-dominated solutions.

Appendix A: Notation

Network:

A	set of links
N	set of nodes
n, m	number of nodes and links
$A(i)$	set of all outgoing links from node i
x_{ij}, y_{ij}	link state on link ij (the subscript ij is ignored if all links have the same set of possible states
$\chi(i,j)$	set of link states (if all $\chi(i,j), \forall ij$ are identical, it is denoted as χ)
w	a successor link (SL) state, defined as $\{\cdots, x_{ij}, \cdots\}, \forall ij \in A(i)$
W_i	set of successor link state at node i
k^{rs}, l^{ks}	path k or l between an OD pair (r,s)
c_{ij}	random link traversal time on link ij
π_{kx}^{rs}	random traversal time on path k^{rs}, given the first outgoing link is in state x
K^{rs}	set of all paths between r and s
Γ_1^{rs}	set of FSD-admissible paths between an OD pair (r,s)
Ω_1^{rs}	set of FSD-optimal paths between an OD pair (r,s)
$k^{js} \Diamond ij$	path k^{js} is extended along link ij
$\succ_1$	first order stochastic dominance

Probability:

ρ_{ijl}^{xy}	transition probability from state x on link ij to state y on link jl
$u_{kx}^{rs}(b)$	cumulative distribution function of π_{kx}^{rs}
$U_w^{rs}(\cdot)$	Pareto frontier function given the successor link state is w at node r
$f_{ijx}^t(\cdot)$	probability density function (pdf) on link ij at state x when departing node i at time t
$P_{ijx}^t(\cdot)$	probability mass function corresponding to $f_{ijx}^t(\cdot)$
$F_{ijx}^t(\cdot)$	cumulative distribution function corresponding to $P_{ijx}^t(\cdot)$ or $f_{ijx}^t(\cdot)$

Algorithm:

ϕ	length of discrete time intervals
T	total time budget
L	number of time intervals in T
Q	candidate list for label correcting algorithm
$\gamma_{\succ}^1(K)$	operator that retrieves FSD-admissible path from a path set K

Acknowledgement This research is partially funded by Federal Highway Administration's UTC program through Northwestern University's Center for the Commercialization of Innovative Transportation Technology. We also would like to thank the four anonymous reviewers for their insightful comments.

References

Andreatta, G. and Romeo, L. (1998). Stochastic shortest paths with recourse. *Networks*, 18, 193–204.

Astarita, V. (1996). A continuous time link model for dynamic network loading based on travel time function. *The Proceedings of the 13th International Symposium on Transportation and Traffic Theory*, 79-103.

Bellman, R.E. (2003). *Dynamic Programming*. Dover Publications, New York.

Carraway, R.L., Morin, T.L, and Moskowitz, H. (1990). Generalized dynamic programming for multicriteria optimization. *European Journal of Operational Research*, 44, 95-104.

Fan, Y., Kalaba, R. and Moore, J. (2005). Arriving on time. *Journal of Optimization Theory and Applications*, 127, 497-513.

Fan, Y., Kalaba, R. and Moore, J. (2005). Shortest paths in stochastic networks with correlated link costs. *Computers and Mathematics with Aplications*, 49, 1549-1564.

Frank, H. (1969). Shortest paths in probabilistic graphs. *Operations Research*, 17(4), 583-599.

Fu, L. (2001). An adaptive routing algorithm for in-vehicle route guidance systems with real-time information. *Transportation Research Part B*, 35(8), 749-765.

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

Gao, S. and Chabini, I. (2006). Optimal routing policy problems in stochastic time-dependent networks. *Transportation Research Part B*, 40(2), 93-122.

Hall, R.W. (1986). The fastest path through a network with random time-dependent travel time. *Transportation Science*, 20, 182-188.

Hansen, P. (1979). Bicriterion path problems. *Multiple Criteria Decision Making Theory and Application*, 109-127.

Henig, M.I. (1985). The shortest path problem with two objective functions. *European Journal of Operational Research*, 25, 281-291.

Levy, H and Hanoch, G. (1970). Relative effectiveness of efficiency criteria for portfolio selection. *Journal of Financial and Quantitative Analysis*, 5, 63-76.

Levy, H. (1992). Stochastic dominance and expected utility: survey and analysis. *Management Science*, 38, 555-592.

Loui, R.P. (1983). Optimal paths in graphs with stochastic or multidimensional weights. *Communications of the ACM*, 26, 670-676.

Miller-Hooks, E.D. (1997). *Optimal Routing in Time-Varying, Stochastic Networks: Algorithms and Implementations*. PhD thesis, Department of Civil Engineering, University of Texas at Austin.

Miller-Hooks, E.D. (2001). Adaptive least-expected time paths in stochastic, time-varying transportation and data networks. *Networks*, 37(1), 35-52.

Miller-Hooks, E.D. and Mahmassani, H.S. (2000). Least expected time paths in stochastic, time-varying transportation networks. *Transportation Science*, 34(2), 198-215.

Miller-Hooks, E.D. and H.S. Mahmassani. (2003). Path comparisons for a priori and time-adaptive decisions in stochastic, time- varying networks. *European Journal of Operational Research*, 146(2), 67-82.

Muller, A. and Stoyan, D. (2002). *Comparison Methods for Stochastic Models and Risks*. John Wiley and Sons, Chichester, UK.

Nie, Y. and Fan, Y (2006). Arriving-on-time problem: discrete algorithm that ensures convergence. *Transportation Research Record*, 1964, 193-200.

Nie, Y. and Wu, X. (2008). The shortest path problem considering on-time arrival probability. *Transportation Research Part B*, Under Revision.

Polychronopoulos, G.H. and Tsitsiklis, J.N. (1996). Stochastic shortest path problems with recourse. *Networks*, 27(2), 133-143.

Provan, J.S. (2003). A polynomial-time algorithm to find shortest paths with recourse. *Networks*, 41(2), 115-125.

Sen, S., Pillai, R., Joshi, S. and Rathi, A. (2001). A mean-variance model for route guidance in advanced traveler information systems. *Transportation Science*, 35, 37-49.

Sivakumar, R. and Batta, R. (1994). The variance-constrained shortest path problem. *Transportation Science*, 28, 309-316.

Waller, S.T. and Ziliaskopoulos, A.K. (2002). On the online shortest path problem with limited arc cost dependencies. *Networks*, 40, 216-227.