Optimal Paths in Dynamic Networks with Dependent Random Link Travel Times

He Huang
Singapore-MIT Alliance for Research and Technology
Future Urban Mobility
S16-04-19
3 Science Drive 2
Singapore 117543
Phone: +65-6601-1636
Email: huanghe@smart.mit.edu

Song Gao
Department of Civil and Environmental Engineering
University of Massachusetts Amherst
214C Marston Hall
130 Natural Resources Road
Amherst, MA 01003
Phone: +1-413-545-2688
Fax: +1-413-545-9569
Email: sgao@engin.umass.edu

Abstract

This paper addresses the problem of finding optimal paths in a network where all link travel times are stochastic and time-dependent, and correlated over time and space. A disutility function of travel time is defined to evaluate the paths, and those with the minimum expected disutility are defined as the optimal paths. Bellman's Principle (Bellman, 1958) is shown to be invalid if the optimality or non-dominance of a path and its sub-paths is defined with respect to the complete set of departure times and joint realizations of link travel time. An exact label-correcting algorithm is designed to find optimal paths based on a new property for which Bellman's Principle holds. The algorithm has exponential worst-case computational complexity. Computational tests are conducted on three types of networks. Although the average running time is exponential, the number of the optimal path candidates is polynomial on two networks and grows exponentially in the third one. Computational results in large networks and analytical results in a small network show that stochastic dependencies affect optimal path finding in a stochastic network, and that the impact is closely related to the levels of correlation and risk attitude.

Keywords Optimal path; Non-dominated path; Correlation; Stochastic dependency; Risk aversion

1 Introduction

Traffic networks are inherently uncertain. Random disruptions like crashes, vehicle breakdown, bad weather, construction and maintenance activities greatly affect the reliability of transportation systems and account for more than half of the congestion in the United States' 439 urban areas (Schrank and Lomax, 2009). In order to model travelers' route choice decisions in a transportation system, a stochastic time-dependent (STD) network is required to capture such uncertainties, where link travel times are time-dependent random variables.

Strong dependencies exist among random link travel times, largely due to traffic flow propagations over time and space. For example, congestion occurs upstream of an accident, and high travel time on the accident link at 8:00 AM will indicate high travel time on an upstream link at 8:10 AM. When a region experiences a heavy thunderstorm, all links affected by the weather will experience delays, and high travel times on highways typically indicate high travel times on arterials. Network stochastic dependencies are required to capture the benefits of real-time information for network routing, since only through the dependencies over time and space can the knowledge of an incident at the current time result in a better prediction of traffic conditions in the future and elsewhere.

In an STD network, the definition of an optimal path may vary, as there exists a large number of optimality criteria (see Section 2 for a detailed review). In this paper, we focus on minimum expected disutility (MED), following the classical von Neumann and Morgenstern paradigm of decision under risk in economics (von Neumann and Morgenstern, 1944). Disutility is defined as an increasing function of travel time that can be either linear or non-linear. The travel time itself can be viewed as a special case of the disutility function. Travelers are assumed to minimize expected disutility when choosing their paths. A brief introduction to the expected utility theory (EUT) for decision under risk is provided.

Consider a prospect $\mathbf{q}$ of risky travel times with their corresponding objective probabilities $\mathbf{q} = \{x_1, p_1; x_2, p_2; \ldots ; x_m, p_m\}$, where travel times x_i are represented by non-negative numbers. A theory for decision under risk assigns a value to a prospect through a preference function $d(\mathbf{q})$ and assumes that a decision maker chooses the prospect with the largest $d(\cdot)$.

The EUT transforms the risky travel time prospect with an increasing disutility function $D(x)$, and assumes $d(\cdot)$ is the mean of the disutility function, namely, $\sum_i D(x_i)p_i$. The risk attitude of a decision maker under EUT is thus completely determined by the curvature of the disutility function, where a convex $D(\cdot)$ suggests risk seeking, concave suggests risk aversion, and affine suggests risk neutrality.

In this paper, we focus on the problem of finding the MED path in an STD network where network stochastic dependencies are incorporated through a joint distribution of all link travel times at all time periods. The contributions of this paper are: 1) Theoretical and computational analyses showing that stochastic dependencies affect optimal path finding in a stochastic network, and that the effect depends on the level of link travel time correlations and travelers' risk aversion; 2) Bellman's Principle (Bellman, 1958) is shown to be invalid if the optimality or non-dominance of a path and its sub-paths is defined with respect to (w.r.t.) the universal set of departure times and travel time probabilistic outcomes; and 3) The introduction of a new property for which Bellman's Principle is valid, and proof that there must exist an optimal path with this property. An exact label-correcting algorithm is designed to find the paths with MED based on this property as an extension of Algorithm EV in Miller-Hooks and Mahmassani (2000).

The paper is organized as follows. Section 2 provides a literature review on optimal path problems under different assumptions of the network. In Section 3, the STD network and the optimal path are defined. A label-correcting algorithm is presented in Section 4, and computational tests are conducted in Section 5. In Section 6, conclusions are made and future directions are proposed.

2 Literature Review

Ever since the early research of Bellman (1958), Dijkstra (1959), and Dantzig (1960), a large number of studies have addressed the problem of finding optimal paths. Different assumptions and constraints have been made in terms of time-dependency of link travel times, randomness of link travel times, and stochastic dependencies among link travel times over time and/or space. In this literature review, the focus is on stochastic networks.

In deterministic networks, algorithms that are similar to the Dijkstra's can be applied in either static cases or first-in-first-out (FIFO) time-dependent cases (Dreyfus, 1969). However, such algorithms are generally not applicable to the optimal path problem in stochastic networks, due to the invalidity of Bellman's Principle of Optimality (Miller-Hooks and Mahmassani, 2000). Moreover, unlike deterministic networks, in which a single optimal path can be determined, a stochastic network may have several paths with positive probabilities of attaining the minimum disutility for some realization of the network. In this case, a set of non-dominated (or Pareto-optimal) paths can be identified.

Several papers have attempted to define the minimum path travel time distribution in static and stochastic networks. Frank (1969) and Mirchandani (1976) have addressed the problem of determining the probability distribution of the minimum path travel time. Frank (1969) assumes continuous probability distributions for link travel times and computes the probability that the minimum path travel time is less than a given threshold. Mirchandani (1976) assumes independent discrete probability distributions for link travel times and develops an algorithm to compute the probability mass function of the minimum path travel time. Sigal et al. (1980) compute the probability that a given path is shorter than all the others, and suggest considering the path with the maximum probability of being the shortest path as optimal.

Minimum expected travel time (METT) and minimum expected disutility (MED) are common optimality criterion. Several works (Loui, 1983; Eiger et al., 1985; Mirchandani and Soroush, 1985; Murthy and Sarkar, 1996; Murthy and Sarkar, 1998) present procedures for finding optimal paths with various forms of disutility functions. It is shown that Bellman's Principle of Optimality is valid when affine or exponential functions are used. More general non-linear disutility functions that capture risk-averse behavior may be approximated by piecewise-linear and convex functions, and Murthy and Sarkar (1998) develop exact algorithms to solve larger instances of the problem.

The METT criterion does not consider the effect of travel time reliability on route choice, while MED with a convex (concave) disutility function models risk aversion (seeking). There are other approaches to considering travel reliability in optimal path finding, such as a bi-criteria shortest path problem that trades off the mean and variance of the path travel time. The bi-criteria problem can be formulated using generalized dynamic programming (Carraway et al., 1990) based on the non-dominance relationship. The mean-variance tradeoff can also be treated in other ways. For example, in Sen et al. (2001), the objective function of stochastic routing becomes a parametric linear combination of mean and variance. Nie and Wu (2009b) employ a label-correcting algorithm to address the problem of finding the shortest paths to guarantee a given probability of arriving on-time.

The optimal path problem is more complex for dynamic and stochastic networks. For example, to find an METT path in a static and stochastic network (with or without stochastic dependency), one can simply set each link travel time random variable to its expected value and solve an equivalent shortest path problem in the converted static and deterministic network. This method, however, cannot be applied in a time-dependent network, as a path travel time is composed of link travel times at the time of arrival of each intermediate node, and the travel time at an "expected arrival time" is generally not the same as the expected travel time over random arrival times. Hall (1986) proposes a branch-and-bound procedure for finding the METT path on this type of network. Miller-Hooks (1997) and Miller-Hooks and Mahmassani (2000) explore the definition of optimality based on first-order stochastic dominance and definite stochastic dominance. Label-correcting algorithms are proposed to find non-dominated paths under the stochastic dominance rules. Recognizing that the exact algorithm does not have a polynomial bound, heuristics are considered to limit the size of the retained non-dominated paths by a predetermined number. However, these heuristic methods may not identify any non-dominated paths, as noted in Miller-Hooks (1997).

Some studies on the optimal path problem take into account network stochastic dependencies. Sivakumar and Batta (1994) discuss the variance-constrained shortest path problem and use covariance matrices to model the correlation across links. Sen et al. (2001) use a similar approach, and assume that removing a cycle results in a route whose total variance is strictly less than that of the route containing the cycle. This cycle covariance assumption allows for a realistic algorithmic approach to real-time implementation and does not rule out negatively correlated link travel times. In Nie and Wu (2009a), travel time correlations are restricted only to adjacent links, and non-dominated paths are generated to find those with maximum arrival time reliability.

A number of works that address the related problem of finding optimal adaptive routing policies have explicitly recognized network stochastic dependencies. We briefly review those works, as the dependency modeling approaches may also have applications in non-adaptive path finding problems. Psaraftis and Tsitsiklis (1993) assume that link travel times are known functions of certain environment variables at network nodes and that each of these variables evolves according to an independent Markov process. Travelers learn the current state of the Markovian chain at any time. The network is assumed to be acyclic to enable the design of a polynomial-time algorithm. Waller and Ziliaskopoulos (2002) examine the adaptive routing problem with limited forms of spatial and temporal link cost dependencies. They assume one-step arc dependence, that is, given the cost of predecessor links, no further information is obtained through spatial dependence. The limited temporal dependency assumes that the cost of a link is known once the entrance node is reached. Fan et al. (2005) address the adaptive routing problem in static and stochastic networks with correlated link service levels. A limited correlation structure which is similar to that in Waller and Ziliaskopoulos (2002) is employed and link states are restricted to either congested or uncongested. Conditional probabilities are introduced to address the correlation between the states of adjacent nodes. They show that the label-correcting algorithm in Waller and Ziliaskopoulos (2002) can also be derived from the dynamic programming point of view. In Boyles (2006), conditional probabilities of adjacent link travel costs are utilized and travelers are assumed to remember only the travel time on the last link they traverse. The objective function is a general piece-wise polynomial function of arrival time at the destination. In a series of one of the co-author's previous studies (Gao and Chabini, 2002; Gao, 2005; Gao and Chabini, 2006; Gao and Huang, 2009; Gao and Huang, 2011), complete dependencies are assumed, where all travel times on all links at all time periods are correlated, and a joint distribution of travel time random variables is applied.

The above review shows that a number of studies have been conducted addressing the optimal path problem in stochastic time-dependent networks with no consideration of stochastic dependencies. There are several studies that examine the optimal path problem in stochastic static networks with consideration of limited dependencies. Little work has been done to examine the problem in networks with a combination of complete stochastic dependencies and time dependencies. This paper fills the gap by presenting an exact algorithm to address the problem.

3 Problem Statement

3.1 The Network

Let $G = (N, A, T, C)$ denote an STD network. N is the set of nodes and A the set of links, with $|N| = n$ and $|A| = m$. There is at most one directional link from node j to k, denoted as (j, k). A path can be denoted as a sequence of consecutive nodes. T is the set of time periods $\{0, 1, \ldots, K-1\}$. Link travel times with entry times between 0 and $K-2$ are time-dependent and random, while those at and beyond $K-1$ are static and deterministic. The time period between 0 and $K-2$ represent the peak hour period, when travel times have higher variability than off-peak hours, which are represented by the time period at and beyond $K-1$. In an STD network with complete dependency, travel times on all links at all time periods are jointly distributed random variables. The travel time on each link (j, k) at each time t is a random variable, which is assumed to be positive and integral, with a finite number of discrete support points. A support point is defined as a distinct value (vector of values) that a discrete random variable (vector) can take. $C = \{C^1, \ldots, C^R\}$ is the set of support points of the joint probability mass function of all link travel times at all times, where C^r is a vector of time-dependent link travel times with a dimension of $K \times m$, $r = 1, 2, \ldots, R$. $C^r_{jk,t}$ is the travel time of link (j, k) at time t in the r-th support point, with probability p_r, and $\sum\limits_{r=1}^{R} p_r = 1$.

3.2 Optimal Path Problem

The optimal path problem is examined from all origins and departure times to a single destination D. $S_\lambda(O, t, r)$ is defined as the travel time of path λ from origin node O and departure time t to the destination node D if support point r is realized. $e_\lambda(O, t)$ is the expected travel time of path λ from origin node O and departure time t to the destination node D where the expectation is taken over all support points. Let $D_\lambda(O, t, r)$ denote the disutility of path λ from origin node O and departure time t to the destination node D in support point r, and $D(\cdot)$ is the disutility function, i.e., $D_\lambda(O, t, r) = D(S_\lambda(O, t, r))$. The disutility function $D(\cdot)$ can be linear or nonlinear, and is an increasing function of travel time. $d_\lambda(O, t)$ is the expected disutility where the expectation is taken over all support points.

$$e_\lambda^{CD}(O,t) = \sum_{r=1}^{R} S_\lambda(O,t,r)\cdot p_r, \tag{1}$$

$$d_\lambda^{CD}(O,t) = \sum_{r=1}^{R} D_\lambda(O,t,r)\cdot p_r. \tag{2}$$

The superscript "CD" stands for "complete dependency", indicating that complete stochastic dependencies among link travel times are considered.

The relationship between the support point travel times / disutilities of a path and of its sub-path is given as follows:

$$S_\lambda(O,t,r) = C_{Ok,t}^r + S_{\lambda'}(k, t + C_{Ok,t}^r, r), \tag{3}$$

$$D_\lambda(O,t,r) = D(C_{Ok,t}^r + S_{\lambda'}(k, t + C_{Ok,t}^r, r)), \tag{4}$$

where node k is the next node on path λ and the starting node of sub-path λ', and $t + C_{Ok,t}^r$ is the exit time out of node k in support point r.

The expected travel time / disutility is then re-written as follows:

$$e_\lambda^{CD}(O,t) = \sum_{r=1}^{R}(C_{Ok,t}^r + S_{\lambda'}(k, t + C_{Ok,t}^r, r))\cdot p_r, \tag{5}$$

$$d_\lambda^{CD}(O,t) = \sum_{r=1}^{R} D(C_{Ok,t}^r + S_{\lambda'}(k, t + C_{Ok,t}^r, r))\cdot p_r. \tag{6}$$

This is different from how the expected travel time / disutility is calculated in an STD network with no stochastic dependencies, where marginal distributions of link travel times are utilized, as shown below:

$$e_\lambda^{ND}(O,t) = \sum_{i=1}^{Q}(C_{Ok,t}^i + e_{\lambda'}^{ND}(k, t + C_{Ok,t}^i))\cdot p_i, \tag{7}$$

$$d_\lambda^{ND}(O,t) = \sum_{i=1}^{Q} D(C_{Ok,t}^i + e_{\lambda'}^{ND}(k, t + C_{Ok,t}^i))\cdot p_i, \tag{8}$$

where the superscript "ND" stands for "no dependency", Q is the number of support points for the marginal distribution of travel time on link (O, k) and p_i the corresponding marginal probability. Note that the equation for $e_\lambda^{ND}(O, t)$ is the same as the equation in Step 2 of Algorithm EV in Miller-Hooks and Mahmassani (2000).

If an exponential disutility function is used to represent risk aversion, i.e., $D_\lambda(O, t, r) = D(S_\lambda(O, t, r)) = \exp(\alpha \cdot S_\lambda(O, t, r))$, the expected disutilities for CD and ND cases are given as follows:

$$\begin{aligned} d_\lambda^{\mathrm{CD}}(O, t) &= \sum_{r=1}^{R} \exp(\alpha \cdot C_{Ok,t}^r) \cdot \exp(\alpha \cdot S_{\lambda'}(k, t + C_{Ok,t}^r, r)) \cdot p_r \\ &= \sum_{r=1}^{R} \exp(\alpha \cdot C_{Ok,t}^r) \cdot D_{\lambda'}(k, t + C_{Ok,t}^r, r) \cdot p_r, \end{aligned} \quad (9)$$

$$d_\lambda^{\mathrm{ND}}(O, t) = \sum_{i=1}^{Q} \exp(\alpha \cdot C_{Ok,t}^i) \cdot \exp(\alpha \cdot e_{\lambda'}^{ND}(k, t + C_{Ok,t}^i)) \cdot p_i. \quad (10)$$

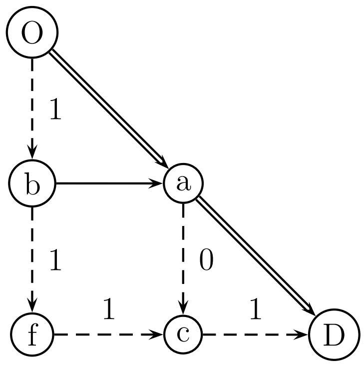
α	0.01	0.1	0.2	0.5	1.0	1.5	2.0	3.0
x	15.1	16.2	17.2	18.6	19.3	19.5	19.7	19.8

Table 1: Traveler's Risk-Averse Attitude

The parameter α in the exponential disutility function represents the level of risk aversion. When α is larger, the traveler is more risk-averse. When α is close to 0, the traveler is close to risk-neutral. Suppose a path has a random travel time of 10 or 20 minutes, each with probability 0.5. Table 1 shows the a value and the corresponding certainty equivalency value x such that a traveler who minimizes the exponential disutility is indifferent between (10, 0.5; 20, 0.5) and (x, 1.0). High-risk paths are equivalent to a poor deterministic value for travelers with a larger α. Therefore, these travelers are less likely to take riskier routes.

The end goal is to determine the MED paths from all origins to a given destination for all departure times. Note that, if the disutility is the travel time itself, we are seeking the paths with METT.

Definition 1 (Path with MED for departure time t) *A path λ with MED from origin O to destination D for departure time t has the minimum expected disutility evaluated over all support points among all the paths between* *the same OD pair and for the same departure time, i.e.,* $\exists$ *no path* λ' *such that* $d_{\lambda'}(O, t) < d_\lambda(O, t)$.



Time	Link	C^1	C^2
	(O, a)	1	2
0	(a, D)	1	M
	(b, a)	1	M
	(O, a)	1	2
1	(a, D)	1	M
	(b, a)	1	1
	(O, a)	1	1
2	(a, D)	1	1
	(b, a)	1	1

Figure 1: The Illustrative Network

An illustrative network is shown in Figure 1 with 6 nodes and 8 links. The travel time on link (a, c) is always 0, and that on any of the other 4 dashed links is 1. Link travel times on solid links are stochastic and time-dependent. There are 2 time periods in the dynamic domain, in which the link travel time random variables are time-dependent ($t = 0$ and 1). There are 2 support points, each with a probability of 1/2, for the joint distribution of 6 travel time random variables on links (O, a), (a, D) and (b, a) over time periods 0 and 1. Travel times at and beyond time 2 are 1 for the 3 links in both support points (static and deterministic). M in the table is a large positive number. For the sake of simplicity, we assume the disutility function is the travel time itself, making this an METT path problem. There are 5 paths from origin O to destination D:

$$\begin{aligned} \lambda_1 &: \quad O \to a \to D; \\ \lambda_2 &: \quad O \to a \to c \to D; \\ \lambda_3 &: \quad O \to b \to a \to D; \\ \lambda_4 &: \quad O \to b \to a \to c \to D; \\ \lambda_5 &: \quad O \to b \to d \to c \to D. \end{aligned}$$

$S_\lambda(O, t, r)$ and $e_\lambda(O, t)$ for each path are shown in Table 2 and the columns under "complete dependency" of Table 3, respectively. Path $\lambda_1(O \to a \to D)$ and path $\lambda_2(O \to a \to c \to D)$ have the METT for all departure times.

Table 2: Path Support Point Travel Time

Path	$C^1, t=0$	$C^2, t=0$	$C^1, t=1$	$C^2, t=1$	$C^1, t \geq 2$	$C^2, t \geq 2$
λ_1	2	3	2	3	2	2
λ_2	2	3	2	3	2	2
λ_3	3	3	3	3	3	3
λ_4	3	3	3	3	3	3
λ_5	4	4	4	4	4	4

Table 3: Path Expected Travel Time

	Complete Dependency			No Dependency		
Path	$t=0$	$t=1$	$t \geq 2$	$t=0$	$t=1$	$t \geq 2$
λ_1	2.5	2.5	2	2.25+M/4	2.5	2
λ_2	2.5	2.5	2	2.5	2.5	2
λ_3	3	3	3	3	3	3
λ_4	3	3	3	3	3	3
λ_5	4	4	4	4	4	4

In general, if we do not consider the stochastic dependency of link travel times, some link travel times that are impossible under certain conditions may be included when calculating expected travel times, which might affect the optimal solution. The columns under "no dependency" of Table 3 show the expected travel time for each path in the same network with the assumption of no stochastic dependency. In this case, each link retains the marginal distribution for travel time as described in Figure 1. There are no joint support points and link travel times are assumed to be independent. For example, in the complete dependency case, if link (O, a) travel time is 1 at time 0, then link (a, D) at time 1 can only have a travel time of 1. However in the no dependency case, travel time on (a, D) at time 1 is assumed to always take its marginal distribution regardless of travel time realizations on other links, and thus can be either 1 or M. This results in a different expected travel time for path $\lambda_1(O \to a \to D)$ as shown in the right half of Table 3.

3.3 Pure Path

In this section, Bellman's Principle of Optimality (Bellman, 1958), defined as the principal that any sub-path of an optimal path must also be optimal, is found to be invalid for the problem context in this paper (Proposition 1). It is also shown that Bellman's Principle of Non-Dominance, defined as the principal that any sub-path of a non-dominated path must also be non-dominated, is also invalid (Proposition 2), even though it is valid in the problems studied by Miller-Hooks and Mahmassani (2000), Opasanon and Miller-Hooks (2006), Miller-Hooks (1997), and Nie and Wu (2009b). We further define a subset of the non-dominated paths as pure paths, and determine that purity is a property that can be maintained across path and sub-path. It is then proved (Theorem 1) that for any origin node, there always exists a pure path with MED, and an exact algorithm can be designed based on this property.

Proposition 1 *A sub-path of a path with MED for a departure time does not necessarily have MED for every possible exit time out of the intermediate node (i.e., the starting node of the sub-path).*

Proof.

We prove this proposition by counterexample. A path with the MED for a departure time has the minimum expectation of disutility evaluated over all support points. However, a sub-path of this path does not necessarily have the MED for all support points for every possible exit time. This sub-path may have a large disutility in some impossible-to-realize support points for some exit times. This large disutility is not included for the calculation of the expected disutility of the MED path, but is accounted for when calculating the expected disutility of the sub-path, making it non-optimal.

In the illustrative network of Figure 1, assuming a simple disutility function of the travel time itself, we can determine that path $\lambda_1(O \to a \to D)$ has the MED for departure time $t = 0$. However, the sub-path $a \to D$ does not have the MED for exit time $t_1 = 1$, since $S_{a\to D}(a, 1, C^2) = C^2_{aD,1} = M$ and $d_{a\to D}(a, 1) = e_{a\to D}(a, 1) = \frac{1+M}{2}$, which is larger than the expected disutility of path $a \to c \to D$ that is a fixed value of 1. Note that, for exit time $t_1 = 1$, C^2 is impossible to be realized if the traveler comes from node O and time 0, i.e., the large travel time M should not be considered in the calculation of the expected travel time from origin O to destination D for departure time 0. **Q.E.D.**

Before defining a non-dominated path, we introduce the complete time-support-point set Ω as the Cartesian product of the sets of time periods T and support points C, that is, $\Omega = \{(t, r)|t \in T, r \in C\}$. Non-dominance is then defined over (a subset of) the universal set Ω.

Definition 2 (Non-Dominated Path) *A path λ from origin O to destination D is non-dominated w.r.t. a subset Ω' of Ω iff $\exists$ no other path λ' between the same OD pair such that*

$D_{\lambda'}(O,t,r) \leq D_{\lambda}(O,t,r), \forall(t,r) \in \Omega'$ *and*

$\exists(t^0,r^0) \in \Omega'$ *such that* $D_{\lambda'}(O,t^0,r^0) < D_{\lambda}(O,t^0,r^0)$.

If not specified, in the remainder of this paper, non-dominance is w.r.t. the complete set of departure time and support points Ω.

In the example from Figure 1, it can be seen from Table 2 that path $\lambda_1(O \to a \to D)$ and path $\lambda_2(O \to a \to c \to D)$ are non-dominated, as for every support point and departure time pair, they have the minimum support point travel time. Note that this is a special case, where non-dominated paths have the same (minimum) support point travel times for all support point and departure time pairs.

A better example can be obtained when evaluating the non-dominated paths from node b to the destination node D. There are three paths: $\mu_1(b \to a \to D)$, $\mu_2(b \to a \to c \to D)$, and $\mu_3(b \to f \to c \to D)$. Table 4 shows the support point travel times for the three paths, and that all three paths are non-dominated.

Table 4: Path Support Point Travel Time

Path	$C^1, t=0$	$C^2, t=0$	$C^1, t=1$	$C^2, t=1$	$C^1, t \geq 2$	$C^2, t \geq 2$
μ_1	2	M+1	2	2	2	2
μ_2	2	M+1	2	2	2	2
μ_3	3	3	3	3	3	3

Because the disutility function is increasing in travel time, non-dominance in terms of disutility is equivalent to non-dominance in terms of travel time. Thus, the $D_{\lambda}(O,t,r)$ terms in Definition 2 can be changed to $S_{\lambda}(O,t,r)$ terms.

It should also be noted that, in an STD network with stochastic dependencies among link travel times, the non-dominance over support points is required in order to take the dependencies into account. In Miller-Hooks and Mahmassani (2000) and Nie and Wu (2009b), the dominance is defined only over time, as they do not consider network stochastic dependency. In the complete dependency case, the travel time on the next link of a path and that on the sub-path are dependent not only through time-dependency of travel times from the next node, but also through stochastic dependencies. It follows that if only expected travel times are used in defining non-dominance, generating non-dominated paths from non-dominated sub-paths could result in the wrong non-dominance set. A similar treatment can be found in Nie and Wu (2009b) where local stochastic dependencies are considered and non-dominance is defined over the states of the outgoing links.

However, even with non-dominance defined over both time and support points, Bellman's Principle still does not apply, as stated formally in the following proposition.

Proposition 2 *A sub-path of a non-dominated path w.r.t. the complete set of departure time and support points* Ω *is not necessarily non-dominated w.r.t.* Ω*.*

Proof.

We prove this proposition by counterexample. The non-dominated path is non-dominated w.r.t. the complete set of departure time and support points Ω. However, it is possible for a sub-path to have an equal disutility as another path for a subset Ω', which is relevant in the composition of the path travel time from the sub-path, but is dominated by that path in other time periods and support points which are irrelevant in the composition. As a result, the sub-path is dominated w.r.t. the complete set.

In Figure 1, the sub-path $a \to D$ of non-dominated path λ_1 has the same travel time as $a \to c \to D$ in support point C^1 for all exit times, but has travel time M for exit time 0 and 1 in support point C^2, and so is dominated by $a \to c \to D$ whose travel time is always 1. Note that this large travel time M cannot be realized if the traveler comes from node O and time 0, i.e., it is not considered in the calculation of the travel time from O to D. **Q.E.D.**

Note that, in Propositions 1 and 2, Bellman's Principle does not hold for the complete set of departure times and support points Ω at the intermediate node. This should not be confused with the fact that it will be valid if the departure time and support point sets are adequately defined at the intermediate node.

The path with the MED for a departure time as defined in this paper has the minimum expected disutility evaluated over all support points. For every possible exit time out of an intermediate node, the sub-path starting from the intermediate node must have the minimum expected disutility evaluated over the compatible support points given the traversal history so far, but does not necessarily achieve the minimum when evaluated over all support points. For example, in the illustrative network of Figure 1, from Table 2, we can determine that path $\lambda_1(O \to a \to D)$ has METT for departure time $t = 0$. There are two possible exit times out of the intermediate node a: $t_1 = 1$, and $t_2 = 2$. For exit time $t_1 = 1$, the corresponding support point is C^1, and the sub-path $a \to D$ has METT for exit time $t_1 = 1$ at C^1; for exit time $t_2 = 2$, the corresponding support point is C^2, and the sub-path $a \to D$ has METT for exit time $t_2 = 2$ at C^2. However as shown before, $a \to D$ does not have METT at time 1 if the expectation is taken over C^1 and C^2.

Similarly, the non-dominated path is non-dominated w.r.t. the complete set of departure time and support points Ω. The sub-path at an intermediate node is non-dominated w.r.t. such a subset Ω' that contains all the possible pairs of the exit time out of the intermediate node and the corresponding support points. For example, in the illustrative network of Figure 1, path $\lambda_1(O \to a \to D)$ is non-dominated w.r.t. Ω. The set of possible exit times out of the intermediate node a and the corresponding support points is $\Omega' = \{(1, C^1), (2, C^1), (2, C^2), (3, C^1), (3, C^2) \cdots\}$. For those possible exit time and corresponding support point pairs, the sub-path $a \to D$ is non-dominated, as the travel time is always 1, the same as (in other words, not dominated by) $a \to c \to D$.

The above observations however cannot build a tractable case. There are potentially 2^{KR} relevant time-support-point set Ω' (the powerset of Ω), and generating a non-dominated path set for each of them is intractable. Fortunately, a property related to non-dominance and termed purity satisfies Bellman's Principle for the complete set Ω.

Definition 3 (Pure Path) *A path is pure iff the path itself and all its sub-paths are non-dominated w.r.t. the complete set of departure time and support points Ω; otherwise, it is a mixed path.*

For the example of Figure 1, path $\lambda_2(O \to a \to c \to D)$ is a pure path from origin O to destination D.

Any pure path is a non-dominated path, while a mixed path can be either dominated or non-dominated. Any dominated path is a mixed path, while a non-dominated path can be either mixed or pure. This relationship can be represented by the following chart, where the outer rectangle represents the complete set of paths between a given OD pair:

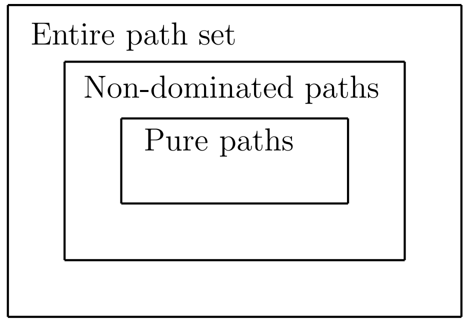


Figure 2: Path Category

Unlike non-dominated paths, pure paths have the property that any sub-path must be pure by definition. The following proposition and theorem guarantee that there must be a pure optimal path.

Proposition 3 *For any mixed path γ from origin node O to destination D, there exists a pure path λ such that $D_\lambda(O,t,r) \leq D_\gamma(O,t,r), \forall(t,r) \in \Omega$.*

Proof.

We prove the proposition by induction.

Basis. When $t = K-1$, link travel times become static and deterministic. Pure paths are optimal, and any mixed path (non-optimal) is dominated by a pure (optimal) path. Therefore Proposition 3 holds.

Inductive step. Suppose Proposition 3 holds at any time $t \geq \tau + 1$. Consider a mixed path γ at $t = \tau$ and node O. If γ is dominated, denote the non-dominated path that dominates γ as η, and η can be either pure or mixed. If γ is non-dominated, set $\eta = \gamma$, and η is mixed non-dominated. Therefore, $D_\eta(O,\tau,r) \leq D_\gamma(O,\tau,r), \forall r$.

Now consider the non-dominated path η.

Case 1: η is pure. Set $\lambda = \eta$, so $D_\lambda(O,\tau,r) \leq D_\gamma(O,\tau,r), \forall r$, and the proof is completed.

Case 2: η is mixed. Denote the next node as k. If the sub-path η' from node k to the destination is mixed, we can replace it with a pure path λ', where $D_{\lambda'}(k, \tau + C^r_{Ok,\tau}, r) \leq D_{\eta'}(k, \tau + C^r_{Ok,\tau}, r), \forall r$ according to the inductive assumption that Proposition 3 holds at any time $t \geq \tau + 1$. Note that $\tau + C^r_{Ok,\tau} \geq \tau + 1$ due to the positive and integer travel time assumption. The disutility function is an increasing function of travel time, so $S_{\lambda'}(k, \tau + C^r_{Ok,\tau}, r) \leq S_{\eta'}(k, \tau + C^r_{Ok,\tau}, r), \forall r$. Then for the resulting path λ:

$$
\begin{aligned}
S_\lambda(O,\tau,r) &= C^r_{Ok,\tau} + S_{\lambda'}(k,\tau + C^r_{Ok,\tau}, r) \\
&\leq C^r_{Ok,\tau} + S_{\eta'}(k,\tau + C^r_{Ok,\tau}, r) \\
&= S_\eta(O,\tau,r), \forall r
\end{aligned}
$$

The disutility function is an increasing function of travel time, so $D_\lambda(O,\tau,r) \leq D_\eta(O,\tau,r) \leq D_\gamma(O,\tau,r), \forall r$.

Since η is non-dominated, λ is also non-dominated. Furthermore, the sub-path of λ is pure, so λ is pure and Proposition 3 is true at time τ and node O.

With the basis and inductive step combined, Proposition 3 holds $\forall(t,r) \in \Omega$. **Q.E.D.**

Note that in the basis step, the proposition also holds in the static time period without the deterministic assumption. In other words, a sub-path of a non-dominated path in a static stochastic network must be non-dominated.

A straightforward conclusion can be drawn that, if a mixed path has the MED for a departure time, then there must exist a pure path with the same MED for the same departure time. This leads to the following theorem:

Theorem 1 (Optimal Pure Path) *For any origin O and departure time t, there exists a pure path with the MED.*

Definition 3 and Theorem 1 show the two most important properties of the pure paths: any sub-path of a pure path must be pure, and it is guaranteed that there is a pure optimal path. Therefore we can construct a pure path based on downstream pure paths, and, as long as we find all pure paths, we can find the pure optimal path(s). Moreover, the set of pure paths is the same for any disutility function as long as it is increasing with travel time, i.e., for any type of users, no matter whether they are risk-averse or risk-seeking, assuming their risk attitudes can be described by the EUT. However, the final optimal path may vary for users with different risk attitudes.

Other properties of the pure paths are given as follows.

From Proposition 3 we draw that, for any mixed non-dominated path γ from origin node O to destination D, there exists a pure path λ such that $D_\lambda(O,t,r) = D_\gamma(O,t,r)$, for all $(t,r) \in \Omega$, i.e., they share the same travel time distribution. However, for a pure path, it is not necessarily true that there exists a mixed non-dominated path that shares the same travel time distribution. If such a path does exist, we term it a shadow path of the pure path. Note that, for any pure path, there may be no shadow path or there may be multiple shadow paths. This is indicative of the relationship between the set of non-dominated paths and the set of pure paths in Figure 2.

When faced with errors in travel time distribution data, a pure path is a more robust routing choice than a shadow path. If travel time data in a support point is incorrect, a traveler might arrive at an intermediate node earlier or later than he/she should according to the data. Under the circumstances, the sub-path of a pure path is still a non-dominated path from the intermediate node to the destination, which is not guaranteed for its shadow path, i.e., a mixed non-dominated path.

4 Algorithm CD-Path

4.1 Solution Approach

Algorithm CD-Path is designed to find all pure paths and thus will find pure paths with the MED for every departure time. However, it does not identify the shadow paths, including those shadow paths with the MED. Note that Algorithm CD-Path finds pure paths using support point travel times rather than support point disutilities due to the equivalence between the non-dominance/purity w.r.t. these two.

The algorithm maintains a set of pure paths for each node j, denoted as $\chi(j)$. A scan eligible (SE) list is used to identify each distinct pure path by the node-path pair (j, λ). At each iteration of the algorithm, a pair (k_0, λ_0) is selected from the SE list. Two pointers are required for each path λ at each predecessor node j to store the pure paths: π_j^λ, indicating the next node; and L_j^λ, indicating the sub-path out of next node. A new path λ is constructed (if not yet) for each possible predecessor node j by making k_0 the next node and λ_0 the sub-path, i.e., $\pi_j^\lambda = k_0$, and $L_j^\lambda = \lambda_0$. λ is added to $\chi(j)$ and dominance among the set is checked. Dominated paths are removed from the set, and temporally non-dominated paths are maintained. Upon termination, the final solution set contains only non-dominated paths.

It should be noted that, at this point, the final solution sets might contain shadow paths, for the reasons that follow. In some iteration, a path λ_0 is added to the pure path set $\chi(k_0)$ and is not dominated by any path in the set in that iteration, so its node-path pair (k_0, λ_0) is added to the SE list, and a path might be constructed for k_0's predecessor nodes based on λ_0, say, λ for node j. In a later iteration, a pure path λ_0' that dominates the mixed path λ_0 is added to the pure path set $\chi(k_0)$ and so the mixed path λ_0 is discarded. At this point, λ at the predecessor node j becomes mixed, yet still stays in the pure set $\chi(j)$. At the end of the algorithm, while λ needs to be explicitly retrieved, it will encounter the problem that its sub-path λ_0 is no longer in the pure path set at node k_0, $\chi(k_0)$. In this case, we can determine that λ is a mixed path and remove it from the pure path set $\chi(j)$. After those mixed non-dominated paths are removed, the final solution set contains only pure paths, and the pure paths with the MED for every departure time can be identified.

Alternatively, the shadow paths can be removed when the sub-path is determined to be dominated. This procedure has the potential advantage of reduced path set size, because mixed paths are removed right away and, by having a smaller current pure path set, a newly generated candidate path at the predecessor node is less likely to be included in the set. However, significant time would be needed to compute all the paths that contain a particular sub-path. Therefore, it is optimal to remove those mixed non-dominated paths at the very end.

Note that, Procedure LR-CHECK is adapted from Nie and Wu (2009b) in order to reduce the amount of effort required to check dominance. We first determine whether the newly generated path λ will update the Pareto frontier, i.e., whether it has a smaller travel time in some support point than the current Pareto frontier. If yes, then λ must be non-dominated, and next we only need to check whether it dominates any path in the current pure path set $\chi(j)$ that does not contribute to the Pareto frontier; if not, then we still need to check whether λ is dominated by any path in $\chi(j)$.

Algorithm CD-Path can be viewed as an extension of Algorithm EV (Miller-Hooks and Mahmassani, 2000). The major difference between the two algorithms is that Algorithm CD-Path works in an STD network where both temporal and spatial dependences are considered while Algorithm EV works in an independent STD network.

As a result, the dominance rule is applied differently for each algorithm. Algorithm CD-Path checks dominance of paths w.r.t. the support point travel times over the complete set of departure time and support point pairs Ω, while in Algorithm EV, the dominance is checked w.r.t. the expected travel times over the departure time set T.

The difference is also reflected in computational demand. The path set is potentially much larger for Algorithm CD-Path, as the chance of being dominated is smaller with a larger dimension for checking dominance. Moreover, longer computation time is potentially required.

Algorithm EV can be extended with Eq. (7) to find the MED path in an independent STD network only if the disutility is either an affine or exponential function of the travel time (Eiger et al., 1985). Only for these two types of disutility functions is the recursive equation between expected disutilities at adjacent nodes valid. In contrast, the non-dominance/purity of path w.r.t. disutility is equivalent to that w.r.t. travel time as long as the disutility function is increasing with travel time, and thus Algorithm CD-Path actually generates all pure paths w.r.t. any increasing disutility function. In other words, Algorithm CD-Path can be applied to any increasing disutility function of the travel time, and is applicable to a wide range of risk attitudes in path finding.

4.2 Algorithm Statement

The steps of Algorithm CD-Path are described next:

Algorithm CD-Path

Step 0. Initialization

Step 0.1. *Initialize labels and path pointers:*

for all $j \in N \setminus \{D\}$ **do**

$$\chi(j) \Leftarrow \phi, \pi_j^c \Leftarrow \infty, L_j^c \Leftarrow \infty,$$
$$S_c(j,t,r) \Leftarrow \infty, e_c(j,t) \Leftarrow \infty, \forall t, \forall r, \forall c \in \{1,2,\ldots,M\}$$

where M is a large enough number so as to permit as many pure paths at any node as might be needed.

end for

for the destination node D:

$$\chi(D) \Leftarrow \{1\}, \pi_D^1 \Leftarrow D, L_D^1 \Leftarrow 1,$$
$$S_1(D,t,r) \Leftarrow 0, e_1(D,t) \Leftarrow 0, \forall t, \forall r$$

Step 0.2. *Initialize the scan eligible list:*

Insert the node-path pair $(D,1)$ in the SE list.

Step 1. Check SE List and Scan Node

if the SE list is not empty **then**

Select the first node-path pair (k_0, λ_0) from the list. Call the associated node k_0 the current node and λ_0 the current path. If the list is empty, go to Step 3.

end if

Step 2: Update Labels

for all $j \in \Gamma^{-1}(k_0)$ (i.e., $\forall j|(j,k_0) \in A$) **do**

Step 2.1. *Temporal Label Creation:*

Set the path pointers: $\pi_j^\lambda \Leftarrow k_0, L_j^\lambda \Leftarrow \lambda_0$, construct a new path λ from j to destination D

Calculate $S_\lambda(j,t,r), \forall t, \forall r$ by Eq. (3):

$$S_\lambda(j,t,r) = C_{jk_0,t}^r + S_{\lambda_0}(k_0, t + C_{jk_0,t}^r, r)$$

Step 2.2. *Label Comparison:*

Add λ to $\chi(j)$ and check dominance among the set. Remove dominated paths from $\chi(j)$. If λ is not dominated by any other path in $\chi(j)$, then add node-path pair (j,λ) to the SE list.

end for

Step 3: Stop and find the Path with MED

For each node j, retrieve each path by recursively combining the next node and next sub-path. If a path is not retrievable due to a missing sub-path, it is a mixed path and discarded. The remaining set $\chi(j)$ contains all pure paths at node j, and the path with the MED can be found for each node j and each departure time t.

Algorithm CD-Path terminates after a finite number of steps, with the set of all pure paths at each node. It has exponential worst-case computational complexity, but the computational tests in Section 5 show that the set of pure paths in a typical transportation network is much smaller than the worst case. Please see Appendix A for three proofs for pure termination sets, finite steps to termination, and exponential worst-case complexity, respectively.

$T \times R$ labels are required for each path as support point travel time is stored at each departure time, resulting in high memory requirements. The computational tests conducted in Section 5 also show that the limit of the computation comes from the memory. If support point travel times are not stored as labels, but calculated each time needed, memory needs can be significantly reduced. However this approach requires prohibitively longer computational time, rendering it practically infeasible.

One potential solution could be a heuristic method that limits the size of the pure path set as a tractable number M (Miller-Hooks and Mahmassani, 2000). However, Miller-Hooks (1997) shows that such a heuristic might not find the optimal path. Masin and Bukchin (2008) propose another algorithm based on the concept of diversity maximization, where the final set includes feasible paths that are as different from each other as possible. Nie et al. (2011) implements the heuristic in an optimal path problem with second-order stochastic dominance. Other heuristic methods include

1) certainty equivalent approximation, which replaces every link travel time random variable by its expected value and thus transforms the stochastic network into a deterministic one; 2) aggregating the distribution, where we check the similarity of the support points, group the similar ones, and replace every link travel time random variable by its expected value within the group, and thus the number of support points is reduced; and 3) working on a limited number of scenarios, e.g., after aggregating the distribution, we can choose a certain number of scenarios such as most-likely scenario, best scenario and worst scenario, and evaluate them only.

It is desirable for us to explore the actual difference between the pure path set and the non-dominated path set. Note that, since non-dominated paths could be mixed paths, i.e., they could contain dominated sub-paths, generating the non-dominated path set would require enumerating all paths. In Section 5 we adapt Algorithm CD-Path to generate non-dominated paths and run tests to investigate the difference.

5 Computational Tests

The objectives of the computational tests are to: 1) investigate the average running time of Algorithm CD-Path as a function of the network size in all three types of networks; 2) investigate the size of the pure path set as a function of network size in all three types of networks; 3) study computationally how the risk aversion coefficient affects the optimal path solution; and 4) study computationally how the level of stochastic dependency affects the optimal path solution.

5.1 Network and Link Travel Time Distribution

The computational tests in this section are conducted in three types of networks: step networks, grid networks, and random networks, which utilize a randomly generated topology. Detailed information on each network type is provided.

5.1.1 Step Network

Theoretically, in an STD network all links have random travel times. However, in order to have a tractable yet realistic model, the most variable part of the network is treated as stochastic and the rest is treated as deterministic.

In this paper, we term the network in Figure 3 a step network. The double-lined links on the diagonal are freeway links, and the nodes on the diagonal are freeway entrances/exits. The horizontal solid link next to each freeway entrance node is an on-ramp link, and the remaining dashed links are local links or off-ramp links. Freeway links and on-ramp links have stochastic and time-dependent travel times, while local links and off-ramp links have static and deterministic travel times.

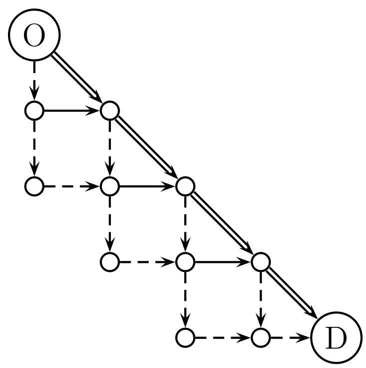


Figure 3: Step Network

A step network can be viewed as a representative transportation network for a typical transportation corridor with a highway and parallel arteries. The underlying rationale is that the variations of the travel times on freeway links are similar and much larger than those of the travel times on local links. The all-local path represents the shortest among all local paths that do not have much variability and can be treated as deterministic, and other all-local paths are removed from the original network. Those deterministic links could be restored to the step network without changing the optimal path solution or the complexity of the problem.

For a step network of level n, there are $3n$ nodes, $n + 1$ of which are freeway exits, and $5n - 2$ links: n freeway links, $n - 1$ on-ramp links and $3n-1$ local links or off-ramp links. The network in Figure 3 is a step network of level 4, and the one in Figure 1 is of level 2. In a step network, there is one all-freeway path and one all-local path. The other paths are mixed with freeway links, on-ramp links, local links and/or off-ramp links.

5.1.2 Grid Network

The grid network is typically seen in planned urban areas, such as Manhattan. In a grid network, all links have similar variability and can be treated as random. Figure 4 gives an example grid network of level 4. For a grid network of level n, there are $(n+1)^2$ nodes and $2n(n+1)$ links.

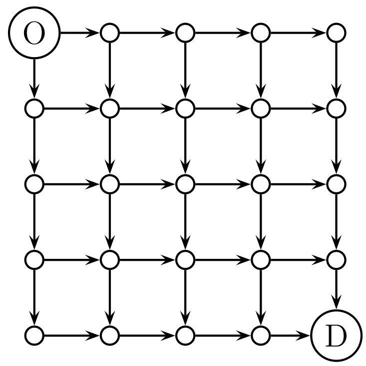


Figure 4: Grid Network

5.1.3 Random Network

The previous network types represent two typical transportation networks, one as a corridor connecting two cities and the other as an urban network. Computational tests are also conducted on networks with randomly generated topology, called random networks in this section. The number of nodes is used as input, the number of links is set as three times the number of nodes, and a random network generator is employed to construct the network topology. More details on the random network generator can be found in Gao (2005) and Gao and Huang (2011).

5.1.4 Link Travel Time Distribution and Other Test Settings

Computational tests are run for all three types of networks.

The tests are conducted on step networks of levels from 3 to 15 (3, 5, 7, 10, 12 and 15). The first freeway node is set as the origin and the last freeway node is set as the destination (the nodes O and D in Figure 3).

The tests on grid networks are conducted for levels from 3 to 7 with 30 time periods. The left-top node is assumed to be the origin and the right-bottom node the destination (the nodes O and D in Figure 4). Note that, although the largest level of the tested grid network is smaller than that of the step network, the number of nodes and the number of links are not actually smaller. For a step network of level 15, the number of nodes is 45 and the number of links is 73; for a grid network of level 7, the number of nodes is 64 and the number of links is 112.

For the purpose of comparison, the number of nodes for random networks is set to be the same as that of the step networks, i.e., the number of nodes range from 9 to 45. The number of links is always three times the number of nodes, i.e., ranging from 27 to 135.

For all three types of networks, the travel times on stochastic links are sampled from truncated (at 3) multivariate normal distribution, with 3 as the original mean, 4 as the original variance, and an original uniform correlation coefficient varying from 0 to 1. Note that the actual mean of the sample is between 4 and 5. The actual variance and the actual correlation coefficient will also differ from the original. The positive uniform correlation coefficient ensures that the covariance matrix is positive semi-definite, and thus is valid. Note that the stochastic links indicate the freeway links and on-ramp links for step network, and all links for grid network and random network. Travel times on deterministic links, i.e., the local links and off-ramp links for step networks, are fixed at 3.

There are 50 support points and 30 time periods for link travel time random variables. For each combination of network level and correlation coefficient, 10 networks are randomly generated. Note that, for the step network and grid network, the network topology remains the same across all 10 while the link travel time distributions vary; for the random network, both are different. The results shown in Section 5.2 are the averages over the 10 networks for each parameter combination.

An exponential disutility function of path travel time is applied, i.e., $D_\lambda(O,t,r) = D(S_\lambda(O,t,r)) = \exp(\alpha \cdot S_\lambda(O,t,r))$, and the expected disutility is given in Eq. (9).

Please find next Table 5 for a summary of the computational test parameters.

5.2 Computational Test Results

Algorithm CD-Path is coded using C++ and tested on a Windows Vista Business (64 bit) workstation with Intel Core i5 CPU 650 @ 3.20GHz and 8.00GB RAM.

Table 5: Summary of the Computational Test Parameters

	Step Network	Grid Network	Random Network
Number of Levels	3, 5, 7, 10, 12, 15	3, 5, 7	N/A
Number of Nodes	9, 15, 21, 30, 36, 45	16, 36, 64	9, 15, 21, 30, 36, 45
Number of Links	13, 23, 33, 48, 58, 73	24, 60, 112	27, 45, 63, 90, 118, 135
Number of Stochastic Links	5, 9, 13, 19, 23, 29	24, 60, 112	27, 45, 63, 90, 118, 135
Number of Support Points	50	50	50
Number of Time Periods	30	30	30
Number of Generated Networks	10	10	10

5.2.1 Running Time and Pure Path Set Size

Tables 6 through 11 show the average running time of Algorithm CD-Path and the average size of the pure path set for all three network types. Note that the algorithm finds optimal paths from all nodes to the destination. For step networks and grid networks, the average size of the pure path set is that of the set for the origin; for random networks, it is the average of the sizes of the sets for all nodes. We present two regressions for each of the six tables: one with the exponential function and the other with the polynomial function. In the regressions, RUN is the average running time over all tested correlation coefficients, SIZE is the average size of the pure path set of the origin node over all tested correlation coefficients, and n is the number of nodes.

The tables show that, for step networks, the average running time of Algorithm CD-Path grows exponentially with the network size and the average size of the pure path set at the origin node seems to grow polynomially with the network size (note that the difference of R^2 for the two regressions is small).

The running time of the algorithm grows exponentially because the algo-

Table 6: Average Running Time vs. Network Size for Step Network

ρ	Step Network					
0	1.145737	1.202073	2.52565	22.17755	105.5225	62.117
0.2	1.165558	1.212483	2.676304	33.05109	57.51884	1285.98
0.4	1.143493	1.207659	2.283864	41.28578	83.39659	1500.43
0.6	1.146104	1.199578	2.116415	27.0273	84.48582	1658.971
0.8	1.143877	1.178022	1.576122	5.609157	20.13407	774.196
1	1.143734	1.146031	1.162972	1.186454	1.380533	1.945746
avg.	1.148084	1.190974	2.056888	21.72289	58.73973	880.6067
$\|N\| = n$	9	15	21	30	36	45
$\|A\| = m$	13	23	33	48	58	73
Level	3	5	7	10	12	15
Regressions	$\text{RUN} = 0.0849 \cdot e^{0.1908n}(R^2 = 0.9409)$					
	$\text{RUN} = 0.00005 \cdot n^{3.9631}(R^2 = 0.7978)$					

Table 7: Average Running Time vs. Network Size for Grid Network

ρ	Grid Network		
0	1.297491	2.360869	215.6773
0.2	1.277201	2.366057	216.1953
0.4	1.277616	2.370289	217.3097
0.6	1.266753	2.372914	220.3326
0.8	1.20332	2.38388	220.7128
1	1.198285	1.29772	3.818742
avg.	1.253444	2.191955	182.3411
$\|N\| = n$	16	36	64
$\|A\| = m$	24	60	112
Level	3	5	7
Regressions	$\text{RUN} = 0.1257 \cdot e^{0.1072n}(R^2 = 0.898)$		
	$\text{RUN} = 0.00005 \cdot n^{3.4018}(R^2 = 0.7542)$		

Table 8: Average Running Time vs. Network Size for Random Network

ρ	Random Network					
0	1.162832	1.199492	1.216785	1.27384	1.319142	1.448924
0.2	1.152093	1.173776	1.200634	1.269578	1.310311	1.456919
0.4	1.1511	1.168702	1.186761	1.265214	1.267603	1.391558
0.6	1.146741	1.162835	1.169498	1.196327	1.210066	1.253251
0.8	1.145219	1.162771	1.163558	1.170083	1.188513	1.178153
1	1.144682	1.146426	1.152655	1.150763	1.155437	1.157254
avg.	1.150445	1.169	1.181649	1.220967	1.241845	1.314343
$\|N\| = n$	9	15	21	30	36	45
$\|A\| = m$	27	45	63	90	118	135
Regressions	$\text{RUN} = 1.1054 \cdot e^{0.0035n}(R^2 = 0.9587)$					
	$\text{RUN} = 0.9596 \cdot n^{0.0747}(R^2 = 0.842)$					

Table 9: Average Size of Pure Path Set vs. Network Size for Step Network

ρ	Step Network					
0	9	56	229.4	662.1	1273.18	643
0.2	9	54	246.4	929.6	923.9	3112.3
0.4	9	52.3	217.3	1106.6	1319	3720.4
0.6	9	49.3	198	859.2	1372.8	3505
0.8	8	35.3	119	295	615.6	1811.6
1	4	6	8	11	13	16
avg.	8	42.15	169.6833	643.9167	919.6833	2134.717
$\|N\| = n$	9	15	21	30	36	45
$\|A\| = m$	13	23	33	48	58	73
Level	3	5	7	10	12	15
Regressions	$\text{SIZE} = 4.0395 \cdot e^{0.1509n}(R^2 = 0.9388)$					
	$\text{SIZE} = 0.0036 \cdot n^{3.507}(R^2 = 0.9969)$					

Table 10: Average Size of Pure Path Set vs. Network Size for Grid Network

ρ	Grid Network		
0	20	252	3432
0.2	20	252	3432
0.4	20	252	3432
0.6	20	252	3432
0.8	20	252	3432
1	8.3	38.7	69.3
avg.	18.05	216.45	2871.55
$\|N\| = n$	16	36	64
$\|A\| = m$	24	60	112
Level	3	5	7
Regressions	SIZE $= 3.8973 \cdot e^{0.1048n} (R^2 = 0.9929)$		
	SIZE $= 0.0007 \cdot n^{3.6179} (R^2 = 0.9881)$		

Table 11: Average Size of Pure Path Set vs. Network Size for Random Network

ρ	Random Network					
0	4.844444	7.726667	7.961905	9.273333	10.15556	11.59556
0.2	4.066667	6.14	7.319048	9.44	9.391667	11.13778
0.4	3.688889	5.793333	6.590476	8.77	8.280556	10.16
0.6	3.555556	4.48	4.695238	5.616667	5.855556	6.711111
0.8	2.522222	3.64	3.319048	3.88	4.225	4.015556
1	1.344444	1.626667	1.633333	1.666667	1.605556	1.564444
avg.	3.337037	4.901111	5.253175	6.441111	6.585648	7.530741
$\|N\| = n$	9	15	21	30	36	45
$\|A\| = m$	27	45	63	90	118	135
Regressions	SIZE $= 3.2507 \cdot e^{0.0202n} (R^2 = 0.881)$					
	SIZE $= 1.236 \cdot n^{0.4775} (R^2 = 0.9701)$					

rithm potentially needs to check all the paths, the number of which grows exponentially with network size. Moreover, although the final pure path set size is polynomial, the sets in the process of label-correcting might contain a lot more paths, which are later determined dominated. We have checked the number of operations of checking dominance, and it increases exponentially with network size. This provides an evidence for the explanation.

The main reason that the pure path set size grows polynomially is that not all paths are potentially pure. Since the on-ramp link travel time distribution is the same as the freeway link travel time distribution, taking off- and then on-ramps is less optimal than traveling directly on the freeway. The on-ramp link travel time is always larger than the travel time on a local link, so taking on- and then off-ramps will result in greater travel time than traveling on two consecutive local links. Therefore, frequently taking on- and off-ramps is not an attractive option and this type of path is not likely to be in the pure set. In summary, only three types of paths are potentially pure: 1) the all-freeway path; 2) the all-local path; and 3) the freeway-local paths with a small number of on-ramp and off-ramp links. For a path of type 3, if the number is one, once the traveler is off the freeway, he/she can never return to the freeway. In that case, the number of paths of type 3 is $O(n^2)$. Similarly, for the freeway-local paths with a small number of on-ramp and off-ramp links (not restricted to be one), the number is polynomial with n.

Another interesting observation is that the pure path set size is relatively small when the correlation is low (e.g., $\rho = 0$) and high (e.g., $\rho = 0.8$ or 1), as is the running time. Path travel time is the sum of link travel times, so its variance increases with link covariance. When the correlation is low (e.g., $\rho = 0$), the variance of path travel time is small, and so the all-freeway path travel time approximately equals the network level (n) times the expected freeway link travel time (between 4 and 5) in every support point, which is smaller than the all-local path travel time ($6n$). Thus, the all-freeway path is more attractive than when the correlation is slightly higher, and so the pure path set size is relatively small. On the other hand, when the correlation is high (e.g., $\rho = 0.8$ or 1), the variance of path travel time is large, and, in this case, taking on- and off-ramps is a poor choice. Thus, paths with a relatively large number of on- and off-ramp links would not be in the pure path set. For $\rho = 1$, the situation becomes extreme. Only all-freeway path, all-local path and those freeway-local paths with consecutive freeway links, then one off-ramp link, and then consecutive local links are pure. The pure path set size is $n + 1$.

In grid networks, the size of the pure path set seems to grow exponentially (again, note that the difference of R^2 for the two regressions is small) as well as the average running time.

For the same network level, a grid network generates many more pure paths than a step network. This is in part because there are more nodes and links, but mainly because in a grid network, most of the paths are similar to each other and one path is not likely to dominate another. It can be observed from Table 10 that, for correlation coefficient $\rho \neq 1$, for each network size, the number of pure paths from the origin to the destination are the same across the correlation values. The number is exactly the total number of paths from the origin to the destination: $(2n)!/(n!n!)$.

Because it is only feasible to run tests on grid networks up to level 7, a heuristic method should be considered to bound the size of the pure path set to a tractable number. As pointed out by Miller-Hooks (1997), such a heuristic might not find the optimal path. However, in a grid network where all paths are relatively similar, a sub-optimal path may be similar enough to an optimal path for these purposes.

In random networks, it seems that the average running time grows exponentially while the size of the pure path set seems to grow polynomially with the network size.

Both the running time and the average pure path set size are extremely small compared with those in the other two types of networks. The reason is that, for a random network whose topology is randomly generated, it is quite possible that there are a small number of relatively short paths (in terms of the number of links in a path) connecting each node to the given destination, which dominate all other paths. It can be observed that the average size of the pure path sets decreases as the correlation coefficient grows. One possible explanation is that, with a larger correlation coefficient, the number of aforementioned relatively short paths is smaller.

5.2.2 Pure Path Set vs. Non-Dominated Path Set

More tests are run to compare the pure path set and the non-dominated path set for all the network types. The algorithm to generate non-dominated paths is similar to Algorithm CD-Path, except that the dominated paths are marked as "dominated" in Step 2.2 rather than discarded.

The tests show that, for all the networks generated, the non-dominated path set is the same as the pure path set, i.e., a shadow path never exists in the tests. We explain the result as follows.

If a pure path has a shadow path (note that both are non-dominated), they must have the same travel time distribution for all departure times at the origin. The sub-paths of the two must have the same (conditional) travel time distributions for all possible arrival times at the intermediate node, where the two sub-paths separate. For all other departure times at the intermediate node, the sub-path of the pure path must have no larger travel time than the sub-path of the shadow path in any possible support point, and a smaller travel time for at least one departure time and support point. In other words, the two sub-paths must be similar for some departure times and support points and quite different for others. Figure 1 shows an example of such a situation, where the sub-path $a \rightarrow D$ is the shadow path of $a \rightarrow c \rightarrow D$. The two sub-paths have the same travel time distribution for time 2, and different for times 0 and 1.

However, the above situation does not happen in the computational tests. This is likely due to the fact that all link travel time random variables are sampled from the same distribution and uniformly correlated with each other. Under such circumstances, whether a path is non-dominated largely depends on the number of links it contains. Non-dominated paths tend to contain the same number of links, and fewer than the dominated paths. As a result, the sub-paths of the non-dominated paths also tend to have the same number of links from the intermediate node where they separate, which results in similar travel time distributions for all departure times from the intermediate node. In such a situation, it is not likely for one sub-path to dominate another to create a shadow path.

5.2.3 Risk Aversion and Correlation

Tests are also conducted to study how the risk aversion coefficient and the level of stochastic dependency affect the optimal path solution with an exponential disutility function. Note that grid networks generate an extremely large number of similar paths that do not dominate each other, random networks generate an extremely small number of relatively short paths that dominate all other paths, and the optimal paths of those two types of networks offer little practical information. Therefore, only step networks are used to investigate the relationship between the optimal path solution and the risk aversion coefficient in the disutility function and the correlation coefficient of the link travel time random variables. The all-freeway path is used as a benchmark and tested for optimal circumstances..

Tables 12 and 13 show the largest value of α with which the all-freeway path has the MED from the origin node to the destination node for a given link correlation coefficient in two cases, one with stochastic dependencies considered (complete dependency) and the other without (no dependency). The range of the tested α values is from 0 to 10 with step 0.01.

Table 12: Largest Value of α for an Optimal All-Freeway Path (Complete Dependency)

	Network Level of Step Network					
ρ	3	5	7	10	12	15
0	10	9.092	1.773	3.625	1.045	0.604
0.2	2.332	9.0174	2.473	0.274	1.619	0.179
0.4	1.29	0.295	0.178	0.158	0.12	0.115
0.6	0.358	0.306	0.131	0.127	0.112	0.087
0.8	0.267	0.135	0.094	0.086	0.076	0.05
1	0.165	0.132	0.113	0.053	0.035	0.059

Table 13: Largest Value of α for an Optimal All-Freeway Path (No Dependency)

	Network Level of Step Network							
ρ	3	5	7	10	15	20	30	50
0	1.93	1.74	1.83	1.84	1.73	1.89	1.81	1.88
0.2	2.01	1.76	1.92	1.77	1.88	1.88	1.83	1.92
0.4	2.13	1.97	2.02	1.87	1.91	1.77	1.85	1.83
0.6	2.02	1.80	1.88	1.86	1.85	1.86	1.80	1.88
0.8	1.94	1.77	1.89	1.76	1.73	1.76	1.95	1.77
1	1.82	1.93	1.98	1.91	1.80	1.74	1.84	1.77

An adapted Algorithm EV (Miller-Hooks and Mahmassani, 2000) is applied to generate optimal paths in the no-dependency case. The expected disutility is calculated based on Eq. (10), which replaces the equation in Step 2 of Algorithm EV. The original Algorithm EV finds the paths with the least expected travel time and thus implicitly assumes risk-neutral users. In order to compare Algorithm CD-Path with Algorithm EV and show the effects of the link travel time correlations and the degree of travelers' risk-averse attitude on the optimal path solution, the original Algorithm EV must be adapted for use with the exponential disutility function. Note that the same network data are used as in the complete dependency case, but that a different algorithm is used to treat link travel times as independent.

It is shown that the all-freeway path is more attractive when the correlation and/or risk aversion is low. In the complete dependency case, the boundary value of α decreases with ρ, suggesting that the all-freeway path is more attractive when the correlation is lower for a given risk aversion level. Furthermore, the boundary value of α decreases with the network size, suggesting that when the network size is larger, the all-freeway path is less likely to be optimal. This is because OD paths in a larger network have a larger number of links and thus the effect of link correlation on path travel time risk is more prominent, which is to the disadvantage of the most risky path – all-freeway path. If the travel times are assumed independent, Table 13 shows that the boundary value of α is virtually independent with the correlation. This is expected as the correlation is used only in the data generation and ignored by the adapted Algorithm EV. This shows that ignoring stochastic dependency would generate the same optimal path regardless of the correlation, yet in reality the optimal path changes with correlation. Comparing the α values in Tables 12 and 13, the difference between the complete dependency case and the no dependency case is small when the correlation is low. When the correlation is high, the complete dependency case shows that the all-freeway path is optimal only with a very small α, while the no dependency case shows the same α values as when the correlation is low.

6 CONCLUSIONS AND FUTURE DIRECTIONS

This paper addresses the optimal path finding problem in a stochastic time-dependent network where all link travel times are temporally and spatially correlated. It is shown that, in such a network, Bellman's Principle does not hold if the optimality or non-dominance is defined w.r.t. the complete set of departure time and support point pairs for the path and its sub-paths. A property related to non-dominance is found to satisfy Bellman's Principle for the complete set, and it is proved that, for any origin node, there always exists a pure path with MED. An exact label-correcting algorithm is designed to find the optimal paths with MED. Computational tests show that the average running time of Algorithm CD-Path grows exponentially with network size, and that the average size of the pure path set grows polynomially in a step network with properly defined stochastic links or a random network, and exponentially in a grid network. Computational tests in large step networks and analytical solutions in a small step network show that the all-freeway path is more attractive when link correlation and/or risk aversion is low. The difference between the complete dependency case and the no dependency case is not significant when the correlation of link travel times is low, and relatively large when the correlation is high.

Additional computational tests on real-life networks would greatly benefit continued understanding of the optimal path problem in stochastic transportation networks. Traffic data could be obtained (e.g., from the PeMS database) and analyzed to study the characteristics of stochastic dependencies among link travel times. Further research plans include the creation of a correlation prediction model using a linear or non-linear regression on the observed data. This model could show how correlation changes over time and space, and provide a more realistic covariance matrix for link travel time random variables.

Further research will also provide insight into the extent of spatial and temporal dependencies. For example, given the incoming link travel times at 8:05 AM, will the knowledge of those further upstream at 8:00 AM provide additional useful information about the outgoing link at 8:05 AM? In other words, is the travel time random variable of the outgoing link independent from those further upstream, given the incoming link travel times? If such conditional independence exists, the stochastic network can be represented through a set of conditional probability distributions, instead of a joint distribution of all link travel times. This will enable both efficient storage of the representation in computer memory and the design of more efficient algorithms.

Acknowledgements

This study is funded by the Department of Transportation through the University of Massachusetts Initiative UTC (University Transportation Center).

References

Bellman, R. (1958). On a routing problem, *Quarterly of Applied Mathematics* **16**(1): 87–90.

Boyles, S. D. (2006). *Reliable routing with recourse in stochastic, time-dependent transportation networks*, Master's thesis, The University of Texas, Austin, TX.

Carraway, R. L., Morin, T. L. and Moskowitz, H. (1990). Generalized dynamic programming for multicriteria optimization, *European Journal of Operational Research* **44**(1): 95–104.

Dantzig, G. B. (1960). On the shortest route through a network, *Management Science* **6**(2): 187–190.

Dijkstra, E. W. (1959). A note on two problems in connection with graphs, *Numerische Mathematik* **1**(1): 269–271.

Dreyfus, S. E. (1969). An appraisal of some shortest-path algorithms, *Operations Research* **17**(3): 395–412.

Eiger, A., Mirchandani, P. B. and Soroush, H. (1985). Path preferences and optimal paths in probabilistic networks, *Transportation Science* **19**(1): 75–84.

Fan, Y. Y., Kalaba, R. E. and Moore, J. E. I. (2005). Shortest paths in stochastic networks with correlated link costs, *Computers and Mathematics with Applications* **49**(9-10): 1549–1564.

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

Gao, S. (2005). *Optimal Adaptive Routing and Traffic Assignment in Stochastic Time-Dependent Networks*, PhD thesis, Massachusetts Institute of Technology, Cambridge, MA.

Gao, S. and Chabini, I. (2002). The best routing policy problem in stochastic time-dependent networks, *Transportation Research Record* **1783**: 188–196.

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

Gao, S. and Huang, H. (2009). Is more information better for routing in an uncertain network?, *The 88th Annual Meeting of Transportation Research Board Compendium of Papers DVD*, Washington, DC.

Gao, S. and Huang, H. (2011). Real-time traveler information for optimal adaptive routing in stochastic time-dependent networks, *Transportation Research Part C* **21**(1): 196–213.

Hadar, J. and Russell, W. R. (1969). Rules for ordering uncertain prospects, *The American Economic Review* **59**(1): 25–34.

Hall, R. W. (1986). The fastest path through a network with random time-dependent travel times, *Transportation Science* **20**(3): 182–188.

Loui, R. P. (1983). Optimal paths in graphs with stochastic or multidimensional weights, *Communications of the ACM* **26**(9): 670–676.

Masin, M. and Bukchin, Y. (2008). Diversity maximization approach for multiobjective optimization, *Operations Research* **56**(2): 411–424.

Miller-Hooks, E. (1997). *Optimal Routing in Time-Varying, Stochastic Networks: Algorithms and Implementation*, PhD thesis, The University of Texas, Austin, TX.

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

Mirchandani, P. B. (1976). Shortest distance and reliability of probabilistic networks, *Computers and Operations Research* **12**(4): 365–381.

Mirchandani, P. B. and Soroush, H. (1985). Optimal paths in probabilistic networks: A case with temporary preferences, *Computers and Operations Research* **3**(4): 347–355.

Murthy, I. and Sarkar, S. (1996). A relaxation-based pruning technique for a class of stochastic shortest path problems, *Transportation Science* **30**(3): 220–236.

Murthy, I. and Sarkar, S. (1998). Stochastic shortest path problems with piecewise linear concave linear functions, *Management Science* **44**(11): 125–136.

Nie, Y. and Wu, X. (2009a). Reliable a priori shortest path problem with limited spatial and temporal dependencies, *Proceedings of the 18th International Symposium on Transportation and Traffic Theory*, Hong Kong, China.

Nie, Y. and Wu, X. (2009b). Shortest path problem considering on-time arrival probability, *Transportation Research Part B* **43**(6): 597–613.

Nie, Y., Wu, X. and Homem-de Melo, T. (2011). Optimal path problems with second-order stochastic dominance constraints, *Networks and Spatial Economics* (Doi: 10.1007/s11067-011-9167-6): 1–27.

Opasanon, S. and Miller-Hooks, E. (2006). Multicriteria adaptive paths in stochastic, time-varying networks, *European Journal of Operational Research* **173**(1): 72–91.

Psaraftis, H. N. and Tsitsiklis, J. N. (1993). Dynamic shortest paths in acyclic networks with markovian arc cost, *Operations Research* **41**(1): 91–101.

Schrank, D. and Lomax, T. (2009). 2009 annual urban mobility report, *Technical report*, Texas Transportation Institute.

Sen, S., Pillai, R., Joshi, S. and Rathi, A. K. (2001). A mean-variance model for route guidance in advanced traveler information systems, *Transportation Science* **35**(1): 37–49.

Sigal, C. E., Pritsker, A. A. B. and Solberg, J. J. (1980). The stochastic shortest route problem, *Operations Research* **28**(5): 1122–1129.

Sivakumar, R. A. and Batta, R. (1994). The variance-constrained shortest path problem, *Transportation Science* **28**(4): 309–316.

von Neumann, J. and Morgenstern, O. (1944). *Theory of Games and Economic Behavior*, Princeton University Press, Princeton, New Jersey.

Waller, S. T. and Ziliaskopoulos, A. K. (2002). On the online shortest path problem with limited arc cost dependencies, *Networks* **40**(4): 216–227.

APPENDICES

A. Properties of Algorithm CD-Path

Proposition 4 *Algorithm CD-Path terminates with the set of all pure paths.*

Proof.

Firstly, a proof is provided to show that, upon termination, for each origin node j, all paths in $\chi(j)$ are pure. This is derived from the path construction principle of the algorithm. In Algorithm CD-Path, the dominated paths and all paths that contain the discarded paths as sub-paths are removed from $\chi(j)$. Thus, no mixed paths can remain in $\chi(j)$.

Next, it is established that all pure paths departing from node j are in $\chi(j)$. Suppose there exists a pure path which is not in $\chi(j)$, then either 1) it is constructed and then discarded at some point, or 2) it is never constructed. Case 1 is not possible because it contradicts the fact that a pure path and all its sub-paths are non-dominated. Case 2 is not possible because if so, either the SE list is not empty, which contradicts to the statement of termination, or the path contains at least one sub-path which is dominated, which contradicts to the definition of a pure path. **Q.E.D.**

Proposition 5 *Algorithm CD-Path terminates after a finite number of steps.*

Proof.

Suppose the algorithm does not terminate after a finite number of steps, then the SE list does not become empty after a finite number of steps, thus, either 1) at least one node-path pair enters the SE list for an infinite number of times, or 2) an infinite number of node-path pairs enter the SE list.

Case 1 is not possible because any node-path pair can enter the SE list at most once when it is constructed and remains in the SE list iff it is determined pure.

Case 2 is not possible because the network is finite, and there are a finite number of time intervals and support points. **Q.E.D.**

Proposition 6 *Algorithm CD-Path has exponential worst-case computational complexity.*

Proof.

It is possible that in the worst case, all paths are pure and thus, stay in the final solution set generated by the algorithm upon termination. Consequently, Algorithm CD-Path, which generates all pure paths, is exponential in worst-case computational complexity. **Q.E.D.**

B. Supplemental Analytical Solutions with Continuous Travel Time Distributions

An example with static and continuously distributed travel times where analytical solutions can be obtained is provided. This analysis complements the computational tests with time-dependent and discrete travel time distributions. As will be shown, similar effects of link travel time correlations and the degree of travelers' risk-averse attitude on optimal path solutions are found, which demonstrates the robustness of the results.

In the network of Figure 1, let freeway and on-ramp link travel times be multivariate normal random variables X_1, X_2, X_3, which represent link travel times on (O, a), (a, D) and (b, a) respectively. Assume they have identical mean μ, variance σ^2 and each pair has an identical correlation coefficient ρ. Their joint distribution is written as $X_1, X_2, X_3 \sim \mathrm{MVN}(\mu, \sigma^2, \rho)$. Local link travel time is fixed at μ. The travel times are static.

The distributions for the travel times of the five paths from origin O to destination D are given as follows:

$$\begin{aligned}
\lambda_1 &: \quad X_1 + X_2 \sim \mathrm{N}(2\mu, 2(1+\rho)\sigma^2); \\
\lambda_2 &: \quad X_1 + 2\mu \sim \mathrm{N}(3\mu, \sigma^2); \\
\lambda_3 &: \quad X_3 + X_2 + \mu \sim \mathrm{N}(3\mu, 2(1+\rho)\sigma^2); \\
\lambda_4 &: \quad X_3 + 3\mu \sim \mathrm{N}(4\mu, \sigma^2); \\
\lambda_5 &: \quad 4\mu.
\end{aligned}$$

Compared to λ_3, λ_1 has the same variance, but a smaller mean; and compared to λ_4, λ_2 has the same variance, but a smaller mean. Therefore λ_3 and λ_4 are first-order dominated by λ_1 and λ_2 respectively, and can be eliminated from further analysis (Hadar and Russell, 1969). Note that in this case, the all-freeway path λ_1 is risky yet short, the all-local path λ_5 is risk-free yet long, and the freeway-and-local path λ_2 has moderate risk and a medium travel time.

The disutility functions for the paths are log-normally distributed and their expected values are given as follows:

$$\begin{aligned}\lambda_1: &\quad e^{\alpha(X_1+X_2)} \sim \text{Log-N}(2\alpha\mu, 2\alpha^2(1+\rho)\sigma^2)\\ &\quad E[e^{\alpha(X_1+X_2)}] = e^{2\alpha\mu+\alpha^2(1+\rho)\sigma^2};\\ \lambda_2: &\quad e^{\alpha(X_1+2\mu)} \sim \text{Log-N}(3\alpha\mu, a^2\sigma^2)\\ &\quad E[e^{\alpha(X_1+2\mu)}] = e^{3\alpha\mu+\alpha^2\sigma^2/2};\\ \lambda_5: &\quad e^{4\alpha\mu}.\end{aligned}$$

If $\mu = 3$ minutes (roughly equivalent to freeway exit spacing of 3 miles at 60 mph), and $\sigma^2 = 2^2 = 4$ minutes^2, then the expected disutilities of the paths are:

$$\begin{aligned}\lambda_1: &\quad e^{\alpha(X_1+X_2)} \sim \text{Log-N}(6\alpha, 8\alpha^2(1+\rho))\\ &\quad E[e^{\alpha(X_1+X_2)}] = e^{6\alpha+4\alpha^2(1+\rho)};\\ \lambda_2: &\quad e^{\alpha(X_1+2\mu)} \sim \text{Log-N}(9\alpha, 4\alpha^2)\\ &\quad E[e^{\alpha(X_1+2\mu)}] = e^{9\alpha+2\alpha^2};\\ \lambda_5: &\quad e^{4\alpha\mu} = e^{12\alpha}.\end{aligned}$$

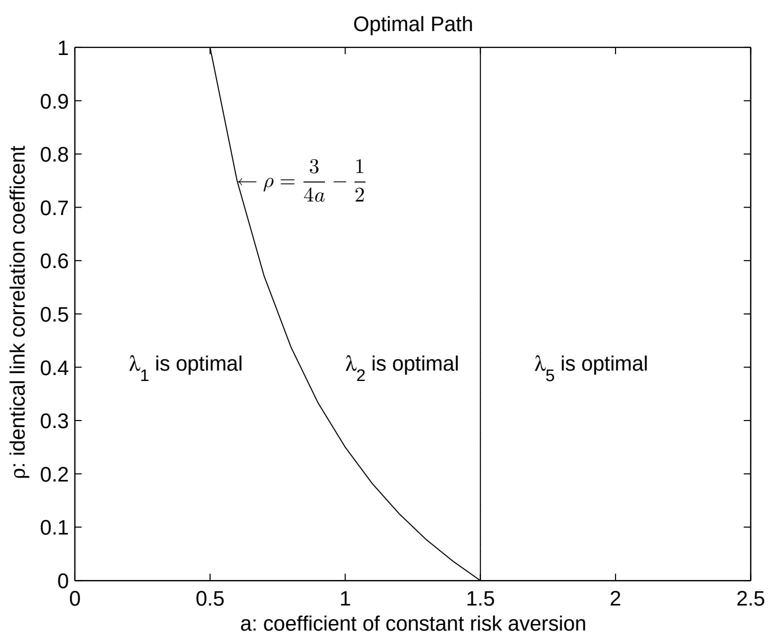


Figure 5: Optimal Path Solution and the Corresponding a and ρ Values

Figure 5 shows how the optimal path solution changes with α and ρ values. The all-freeway path is more likely to be optimal when α is smaller (suggesting an attitude closer to risk neutrality) and ρ is smaller, similar to the results from the computational tests in Section 5.2.

Specifically, when $\alpha < 0.5$, the all-freeway path λ_1 is optimal regardless of the correlation. When $0.5 < \alpha < 1.5$, the all-freeway path λ_1 is optimal for low correlations, and the freeway-local path λ_2 is optimal for high correlations. The boundary dividing "low" and "high" correlations changes with α as specified by the equation in Figure 5. Note that the boundary is derived numerically in the computational tests. When $\alpha > 1.5$, the all-local path λ_5 is optimal regardless of the correlation.

For normally distributed variables, independence is equivalent to zero correlation coefficient. If the stochastic dependencies are ignored as in most existing studies, the horizontal line in Figure 5 with $\rho = 0$ shows that the freeway-local path can never be an optimal path regardless of the risk aversion level, and the all-freeway path is always optimal for $\alpha < 1.5$, which can be viewed as a reasonable range for an average person's risk aversion parameter. This is due to the underestimation of the all-freeway path risk by assuming stochastic independence between links.

C. Glossary

Support point:

A distinct value (vector of values) that a discrete random variable (vector) can take. $C = \{C^1, \ldots, C^R\}$ is the set of support points of the joint probability mass function of all link travel times at all time periods, where C^r is a vector of time-dependent link travel times with a dimension of $K \times m$ (K is the number of time periods and m the number of links), $r = 1, 2, \ldots, R$. $C^r_{jk,t}$ is the travel time of link (j, k) at time t in the r-th support point, with probability p_r, and $\sum_{r=1}^{R} p_r = 1$.

Support point disutility $D_\lambda(O, t, r)$:

The disutility of path λ from origin node O and departure time t to the destination node D in support point r. If the disutility is travel time itself, then $D_\lambda(O, t, r) = S_\lambda(O, t, r)$, which is support point travel time.

Expected disutility $d_\lambda(O, t)$:

The expected disutility of path λ from origin node O and departure time t to the destination node D. If the disutility is travel time itself, then $d_\lambda(O,t) = e_\lambda(O,t)$, which is expected travel time.

Path with MED for departure time t:

A path λ with MED from origin O to destination D for departure time t has the minimum expected disutility over all support points among all the paths between the same OD pair and for the same departure time, i.e., $\exists$ no path λ' such that $d_{\lambda'}(O,t) < d_\lambda(O,t)$.

The complete time-support-point set Ω:

The Cartesian product of the sets of time periods T and support points C, that is, $\Omega = \{(t,r)|t \in T, r \in C\}$.

Non-Dominated Path:

A path λ from origin O to destination D is non-dominated w.r.t. a subset Ω' of Ω iff $\exists$ no other path λ' between the same OD pair such that

$D_{\lambda'}(O,t,r) \leq D_\lambda(O,t,r), \forall (t,r) \in \Omega'$ and

$\exists (t^0, r^0) \in \Omega'$ such that $D_{\lambda'}(O,t^0,r^0) < D_\lambda(O,t^0,r^0)$.

If not specified, non-dominance is w.r.t. the complete set of departure time and support points Ω.

Pure Path:

A path is pure iff the path itself and all its sub-paths are non-dominated w.r.t. the complete set of departure time and support points Ω; otherwise, it is a mixed path.

Risk aversion factor a:

The parameter in the disutility function of travel time which represents the traveler's risk aversion. In the exponential disutility function $exp(aX)$, the value of α shows the level of the traveler's risk aversion. A larger α is associated with a higher level of risk aversion.

Identical correlation coefficient ρ:

In the computational tests, we assume identical correlation coefficient between any pair of link travel time random variables. In other words, in the correlation matrix of link travel time random variables, all non-diagonal elements are the same.