ONLINE STOCHASTIC ROUTING INCORPORATING REAL-TIME TRAFFIC INFORMATION

Lili Du*
Department of Civil, Architectural, and Environmental Engineering
Illinois Institute of Technology
3201 South Dearborn Street
Chicago, IL 60616
USA

Telephone: (312) 567-3426
FAX: (312) 567-3519
lilidu@iit.edu

Srinivas Peeta
School of Civil Engineering
Purdue University, USA
peeta@purdue.edu

Yong Hoon Kim
School of Civil Engineering
Purdue University, USA
kim523@purdue.edu

*Corresponding author
ABSTRACT

This study develops on-line stochastic routing policies which identify the optimal next (path choice) action at the current decision node (intersection) for travelers, based on their preferring future paths with the shortest travel time, the lowest travel time variability, or a combination thereof, given the current network conditions. A modified label-correcting algorithm is provided to solve for the shortest path resulting from the proposed routing policies. Its running time is bounded by $O(mn^2)$, where $m$ and $n$ are the number of arcs and nodes, respectively, in the network. Considering that real-time traffic information is usually available with a certain level of accuracy, the proposed on-line routing policy integrates an existing information fusion model by the authors (1), which provides real-time short-term arc travel time distributions by considering information accuracy. Numerical experiments are used to demonstrate the performance of the proposed routing policies/algorithms as well as the impacts of real-time information accuracy on the online stochastic routing.

**Key words**: online stochastic routing, short-term arc travel time distribution.
Online Stochastic Routing Incorporating Real-Time Traffic Information

INTRODUCTION

In the context of Intelligent Transportation Systems (ITS), various traffic sensors such as loop detectors, probe vehicles and video surveillance systems have been implemented to collect real-time traffic information over the associated road networks. In conjunction with sophisticated data fusion and computing techniques, they can potentially enable stochastic real-time travel time information to be available to travelers. This motivates the development of online stochastic routing algorithms for traveler route guidance systems. In this context, this research proposes online routing policies which seek to find the “best” path for a traveler by taking into account his/her preferences related to travel time and travel time variability in a time-dependent traffic network, given that stochastic real-time travel time information is available to travelers en-route. More precisely, the study considers the following decision-making scenario: a traveler will drive through a network from a given origin to a destination. He/she is provided with real-time travel time information and is guided using a routing policy which identifies the optimal next arc at each intersection based on information derived from a local network. This information network is composed of the road arcs associated with the short-term arc travel time distributions (1) that are being continuously updated with real-time traffic information through an embedded information fusion model.

In the literature, this research is in the domain of the shortest path problem for stochastic time-dependent networks, which allows the arc cost to be a random variable whose probability density function (PDF) could vary with time. Waller and Ziliaskopoulos (2) categorize it into two classes. The first class seeks to compute an a priori solution that optimizes a certain objective function. Efficient Dijkstra-type algorithms and various shortest path algorithms based on Dijkstra's algorithm have been developed (2, 3, 4, 5) for it. The proposed study belongs to the second class, which aims to find an online routing solution that allows multi-stage real-time decision-making. As this topic has been addressed since the 1980s, we first review the literature hereafter and then articulate the contributions of the proposed study.

Hall (6) identifies that when a stochastic network is also time-dependent, a Dijkstra-type algorithm could fail to find a shortest path, and that the optimal solution is not a single path but a routing policy. This concept motivated several studies seeking to generate shortest paths in time-dependent stochastic networks from various perspectives. Wellman et al. (7) identify a qualitative monotonicity condition (stochastic consistency), which enables standard dynamic programming to be applied for developing path planning algorithms in stochastic time-dependent networks. But, the proposed stochastic consistency condition implies that the probability of any given ultimate arrival time at the destination cannot be improved by leaving later, although it may reduce the time duration for traversing an arc. Hence, it is more meaningful for public transit modes such as bus or train rather than the private auto mode which is the focus of the traveler route guidance systems related aspects addressed in this study. Hooks and Mahmassani (8, 9), and Hooks (10) develop algorithms for the least possible cost path, the least expected cost path, and the adaptive least expected cost path for stochastic time-dependent networks, by assuming the arc travel time to be a random variable relative to the arrival time. That is, the possible arrival time at a current node is pre-determined by the possible combinations of the travel time states on the arcs which constitute the path from the origin to this node. By contrast, the proposed study models the arc travel time as a random variable with a time-dependent discrete distribution. This implies that the number of possible arrival times is uncertain and is not limited by the arc travel time states.
which potentially change in real time due to the dynamics of the traffic flow. It is also more consistent in terms of the representation of the real-time traffic information rather than a single value as this information is assumed to be provided by fusing the traffic data from multiple traffic sensors. Fan et al. (11) seek to maximize the probability of arriving at a destination on time in a stochastic time-dependent network. They propose a multistage decision process to determine the optimal action in each stage, but assume the arc travel time to be a stationary independent distribution rather than as a time-dependent distribution (as considered in the proposed study) which is more suitable for developing real-time route guidance strategies. The most closely related work to the proposed study is by Fu (12). It proposes a closed-loop routing policy, also labeled as adaptive routing, where the routing algorithm produces as output a policy instead of single path. It considers the arc travel time as a continuous time stochastic process, and proposes a heuristic algorithm to obtain the expected shortest path in terms of travel time under the developed routing policy, based on the approximations of both the mean and the variance of travel time on a given path.

The proposed study uses the closed-loop routing policy as a conceptual starting point to develop stochastic online routing guidance for travelers en-route given the stochastic time-dependent arc travel time distributions (1) in a dynamic traffic network. It differs from the Fu (12) in two key aspects. First, the study proposes routing policies that integrate the traveler preferences in terms of travel time and travel time variability into the decision process. It acknowledges that travelers have different attitudes to the risk resulting from travel time variability unlike (12) which assumes homogeneity in this context. Second, instead of applying a continuous stochastic process, this study adopts a discrete distribution updated in real-time to describe the dynamic characteristics of the arc travel time. This has been proved to be a more accurate representation of real-time traffic conditions in the field (13). Furthermore, by taking advantage of the discrete arc travel time distribution, this study develops an exact algorithm to find the expected shortest path under the proposed routing policies.

Another important feature of this study is that real-time traffic information is used to determine the optimal remaining path (to the destination) at each intersection en-route. From this viewpoint, the study is close to the shortest path problem with recourse (SSPR) which was first identified by Croucher (3), and further proved to be a NP-hard problem by Provan (14). This branch of work typically considers the influence of the local/partial network information on travelers’ route choices. Croucher (3) assumes that a driver could deviate from the pre-selected arc with a fixed probability based on the availability of new information. Several extensions of this work have been published in the literature (15, 16). Waller and Ziliaskopoulos (2) address the case that the trip decision-maker has the opportunity to re-evaluate the remaining path according to available en-route information such as the deterministic arc cost dependencies in spatial or temporal contexts. Gao and Chabini (17) provide a comprehensive taxonomy and discussion of the variants of the optimal routing policy problems in stochastic time-dependent networks. In addition, they present a general framework for optimal routing problems in stochastic time-dependent networks under the assumption that perfect information is available. Most of the aforementioned studies on SSPR use an implicit method (such as the probability of the deviation from the pre-selected path) to model the influence of real-time information on decision-making en-route. This may lead to some implementation difficulties since it is hard to calibrate such implicit stochastic factors in practice. This study explicitly factors the real-time information into the routing policy/algorithm through an information fusion model. That is, real-time traffic information updates the short-term arc travel time distribution in each time interval, which impacts the solution of the routing algorithm and may potentially re-direct the traveler away from the current path.

Other studies focus on more practical aspects in this problem context. Kim et al. (18, 19) propose a systematic methodology to implement ITS technologies by integrating real-time traffic information.
Their core model of the traffic network is a stochastic time-dependent network with each arc having only two states and the system dynamics being described through an underlying Markov chain model. By contrast, the proposed study models arc travel time as a discrete short-term travel time distribution, which includes comprehensive arc traffic states rather than only two traffic states, thus enabling the generation of more accurate routing solutions. Further, this time-dependent arc travel time distribution is generated by applying an existing data fusion model developed in (1), which takes into account the information quality. Thereby, the proposed routing framework has the capability to include more traffic realism for enabling practical deployment.

In summary, this study develops closed-loop routing policies and the corresponding routing algorithms, which provide online stochastic routing guidance for individual traveler with access to real-time traffic information provided by an advanced route guidance system. The proposed approach differs from the existing efforts in three key aspects. First, traveler preferences relative to travel time uncertainty are explicitly integrated into the routing policies and the corresponding routing algorithms. Second, the arc travel time is described by a discrete travel time distribution, which is updated using real-time traffic information through an existing data fusion model developed in (1). Accordingly, the optimal path en-route is adaptively updated using the latest traffic information. It implies that the influence of real-time traffic information on travelers’ route decision is explicitly considered in the proposed approach. Third, the consideration of information quality in the embedded data fusion model (1) for the determination of the discrete real-time arc travel time distribution assures that the proposed approach entails realism for practical deployment.

The remainder of the paper is organized as follows. The next section presents a preliminary description of the problem. This is followed by the development of the routing policies and the routing algorithm design. Then, the numerical experiments are described and the associated results are discussed. Finally, some concluding comments are presented and potential future work is discussed.

PRELIMINARIES

The proposed approach considers the process of an individual traveler completing an origin-destination (O-D) trip as involving multiple decision-making steps based on the real-time travel time information and the dynamic network topology. This statement has three-fold implications. First, we consider that the network topology associated with the trip changes dynamically. That is, as the trip moves closer to the final destination, the network associated with the current origin (i.e. the intersection) and the final destination becomes smaller since it is not necessary to include the arcs that the traveler has already passed by to decide the future route. Second, the traffic conditions associated with the dynamic network are revealed in real-time. More precisely, we assume that the real-time traffic information accessible to individual travelers keeps updating the short-term arc travel time distribution in discrete time intervals, which is further used to identify the traveler route choice decision at each intersection. Third, the traveler will reevaluate the optimal path at each decision node based on current traffic conditions; this is a possibility under vehicle-to-vehicle communications based traffic system scenarios. Thereby, the completion of a trip is a process with multiple decision-making steps.

The dynamic road network at each decision node at time \( t (= 1, 2, ..., T) \) is represented by a directional acyclic graph \( G^t(N, A) \), where the nodes correspond to intersections (decision nodes) and the arcs correspond to road arcs. Let \( N \) be the set of all nodes (\(|N| = n\)) and \( A \) be the set of all arcs (\(|A| = m\)). \( o \) and \( s \) are used to denote the origin node and the final destination node in \( G^t \). The directed arc from node \( v \) to node \( w \) is denoted by \((v, w)\). The set of arcs incident from node \( v \) is denoted by \( \delta^-(v) \), while the set of arcs entering node \( v \) is denoted by \( \delta^+(v) \). For simplicity in articulation, we do not include the
Routing Policy

A routing policy is an identification of the optimal next action of an individual traveler given the network conditions at the current time. However, travelers may have different attitudes to travel time uncertainty due to departure time, their trip objectives and behavioral tendencies. Thereby, different travelers may make different route choice decisions even when they are provided the same real-time traffic information. For example, if the objective of a trip is to arrive at an airport on time, but the lead time associated with the departure time and the airline boarding time is limited, then most likely the traveler will prefer a route with relatively lower variability. However, if the trip is between the home and a shopping center during the weekend, the traveler may take a certain level of risk and choose the opportunistic shortest path in terms of travel time. By considering such traveler behavior variability, the study develops online routing policies under the closed-loop routing rule by assuming that the short-term discrete arc travel time distribution $\psi(c_a(t))$ is obtained through the information fusion model on each arc.

Routing Policy I seeks the shortest path in terms of the expected minimum travel time ($E^T$) for risk-willing travelers. The mathematical model is shown as Equation (2):

$$a^*(t) = \arg\min_a \{E^T = E[c_a(t)] + c_{is}(t), \forall a = (v, i) \in \delta^-(v)\}$$

where $l_a(t)$ is the total number of states in the discrete travel time distribution on arc $a$ at time $t$. Let $I = \max \{l_a, a \in A\}$. Real-time information is available at discrete time instants with an information accuracy represented by $p^r(t)$, a conditional probability that the real-time information is a true reflection of the actual travel time at time $t$. $\psi(c_a(t))$, referred as to the arc short-term arc travel time distribution in this study, is assumed to be updated using the real-time arc travel time information (when available) through the information fusion model developed by Du et al. (1). The main concept of this information fusion model (1) is briefly summarized hereafter. The information fusion model starts with the past arc travel time distribution available currently in the format illustrated in Equation (1). Whenever real-time information is available, the information fusion model analytically integrates this past arc travel time distribution with the real-time arc travel time information, as part of an adaptive process to dynamically update the short-term arc travel time distribution. In each update process, a nonlinear programming formulation is applied to determine the weights used to integrate the past arc travel time distribution and the real-time information, so that the likelihood of the occurrence of travel time states not reinforced by the real-time travel time information is reduced in the predicted short-term travel time distribution. That is, these weights are determined so as to minimize the uncertainty associated with determining the actual short-term arc travel time distribution.

It is pertinent to note here that the short-term arc travel time distribution and the dynamic traffic network may not be updated synchronously. But, by assuming that the traveler makes a route decision at an intersection based on the latest arc travel time distribution information, this study adopts the same time index to label the dynamic network topology and the short-term arc travel time distribution.

ONLINE STOCHASTIC ROUTING
where, \( a^*(t) \) represents the optimal arc suggested to the traveler at time \( t \) so that under the routing policy he/she will experience the expected minimum travel time from the current node \( v \) to the destination. \( E(c_a(t)) \) is the expected realized travel time on arc \( a \) at time \( t \), \( c_{is} \) is the expected minimum travel time from node \( i \) to the destination \( s \) given that the traveler is routed under the provided policy. It can be expressed as Equation (3):
\[
c_{is}(t) = E\left[ \min \{c_a(t) + c_{js} \mid \forall a = (i,j) \in \delta^-(i) \} \right].
\]
Therefore, Equation (4) also holds.
\[
c_{ss}(t) = 0
\]
Routing Policy II suggests a path with the expected minimum travel time variability \( (E^V) \) for the risk-averse travelers from the current node to the destination. The routing policy is defined by Equation (5):
\[
a^*(t) = \arg\min\{E^V = E(var[c_{is}(t)]) + var(c_{is}(t)), \forall a = (v, i) \in \delta^-(v)\},
\]
where, \( a^*(t) \) represents the optimal arc that the traveler is suggested at time \( t \) so that he/she will experience the path with an expected minimum travel time variance from the current node \( v \) to the destination.
\[
var(c_{is}(t)) = E\left[ \min \{var(c_a(t)) + var(c_{js}(t)) \mid \forall a = (i,j) \in \delta^-(i) \} \right].
\]
Similarly, we have Equation (7):
\[
var(c_{ss}(t)) = 0.
\]
Equations (2), (3), and (4) (or Equations (5), (6), (7)) are formulated as recurrent relations, which can be solved using dynamic programming (20). The corresponding algorithms are designed in next section.

The above routing policies explore two strategies which make routing decisions only considering travel time or travel time variability. Combining these two polices can generate various routing polices, say Routing Policy \( K \), which address route decision-making under more general situations whereby the traveler may systematically balance travel time and travel time variability in their decision-making or prefer to choose the path with the \( k^{th} \) least travel time variability. Integrating the customized weights \( \theta \) to travel time and travel time variance, Equation (8) defines the optimal arc under Routing Policy \( K \):
\[
a^*(t) = \arg\min\{\theta E^T + (1 - \theta)E^V, \forall a = (v, i) \in \delta^-(v)\}.
\]
The question of how to strategically determine the customized weights for travel time and travel time variance represents a future research topic, and is not the focus of the proposed study. For completeness, we demonstrate the effect of \( \theta \) through numerical experiments.

Note that all three proposed routing policies search for the optimal next arc rather than a specific shortest path at each decision node/intersection, based on the latest arc travel time distributions available through the data fusion model in the proposed routing algorithm. Hence, as a traveler arrives to the next decision node, the next optimal arc from this decision node will be determined based on the latest arc travel time distributions at that time updated using available real-time information through the embedded data fusion model. Therefore, the proposed routing policies work on a time-dependent traffic network, and the (latest) arc travel time distributions at the time of the arrival of the traveler to a decision node are incorporated in the proposed routing policies and algorithms.

**Routing Algorithm**

A modified label-correcting routing algorithm is designed to identify the optimal next arc under the proposed routing policies at each decision node. All optimal arcs together constitute the shortest path for
a given trip under a stochastic traffic environment. The optimal arc at each decision node is obtained through the following three steps.

First, the directional network associated with the origin-destination (O-D) of a trip is identified. As the trip evolves, the topology of the network is truncated by removing the previous decision nodes \(v\) and the arcs in \(\delta^-(v)\). This process shrinks the size of the directional network and correspondingly reduces the computational load. In addition, the index of the nodes in the network is sorted by the Topological Order algorithm (21). This step improves the search efficiency of the proposed routing algorithm by avoiding checking of individual nodes more than once.

Next, the algorithm updates the label of each node by the label-correcting algorithm in a reverse order of the node index. The label of each node, denoted by \(\pi_v = E[min\{c_a(t) + \pi_w\} | a = (v, w) \in \delta^-(v)]\) (i.e. Equation (2)), represents the expected minimum travel time from the current node \(i\) to the destination \(s\). To calculate \(\pi_v\), this study recognizes that \(\pi_w\) is a permanent label when we calculate the label \(\pi_v\) since the label-correcting algorithm proceeds in a backward manner. Further, as \(c_a(t)\) is a discrete stochastic variable with limited number of states, there are only a limited number of possible values for \(\pi_v\). We identify all possible values of \(\pi_v\) and the corresponding probability as \(\varphi = \{\varphi_a, p_a\}\), where \(\varphi_a = c_a(t) + \pi_w\forall a = (v, w) \in \delta^-(v)\), and \(p_a\) is the corresponding probability of \(\varphi_a\). The sorted \(\varphi\) is defined as \(\varphi'\). We further find that if \(\varphi_a\) (a value of \(\pi_v\) resulting from the path going through arc \(a' \in \delta^-(v)\)) are listed behind all possible values of \(\varphi_a\), then all values in \(\varphi'\) after \(\gamma\) cannot be the possible value in \(min\{\varphi\}\), where \(\gamma\) indicates the position of the largest value in \(\varphi'\) indexed by arc \(a\). Hence, the proposed sub-algorithm embedded in the label-correcting algorithm calculates the label of node \(v\) using the following steps: (i) sort the possible values in \(\varphi\) to obtain \(\varphi'\); (ii) remove the values from \(\varphi'\) if they cannot appear in \(min\{\varphi\}\) (the final list is denoted as \(\Gamma\)); (iii) calculate the corresponding probabilities of the values in \(\Gamma\); and (iv) calculate \(\pi_v\), the expected minimum travel cost from node \(v\) to node \(s\) under the routing policy.

Finally, the algorithm selects the optimal arc \(a^*\) using Equation (2). Note that in solving the shortest path under Policy I, the means of the travel time states, which are intervals in the short-term arc travel time distribution, are used. This enables the corresponding routing problem to be solved since it is hard to compare the travel time between arcs if their travel time is represented through intervals. The core of the proposed algorithm falls in the framework of label-correcting algorithms. Its computational complexity analysis is provided hereafter.

Proposition: The computational complexity of the proposed algorithm is bounded by \(O(m(\ln)^2)\); if \(I \ll m\) (or \(I \ll n\)), then the computational complexity is reduced to \(O(mn^2)\).

Proof: To analyze the complexity of the algorithm, we use the following logic. Our algorithm is a modified label-correcting algorithm embedded with a sub-algorithm which calculates the expected minimum travel cost from the given node to the destination under our policy. The computational complexity of the label-correcting algorithm with topological order for an acyclic network is \(O(m)\) (21).

Next, we analyze the computational complexity of the embedded sub-algorithm. The first step of the sub-algorithm is to sort the states of the arcs emanating from node \(v\). The number of arcs incident from node \(v\) is bounded by \((n-1)\). Suppose each arc has \(I\) states, the corresponding number of states is bounded by \((In)\). If a Quick Sort algorithm is applied, then the computational complexity of this step is \(((\ln n)\log(\ln n)))\).

Another time-consuming component in the sub-algorithm is to find all possible values of \(min\{\varphi\}\) from the sorted states in \(\varphi'\). In the worst case, the algorithm needs to perform \(I(n-1) - 1\), \(I(n-1) - 2\), \(I(n-1) - 3\), ..., 1 times comparisons and delete only one value from the list. Then, the
corresponding computational load is bounded by \((l(n - 1))^2\). Further, the total computational load of the proposed algorithm is bounded by \(O(m(\log(n) + (n))^2) = O(m(n)^2)\). As this algorithm is designed for real-time applications, and the maximum number of traffic states \(l\) on an individual arc is likely much smaller than the number of arcs (or nodes) under consideration, that is, \(l \ll m\) (or \(l \ll n\)), the computational complexity of the algorithm reduces to \(O(mn^2)\). Then, the proposed algorithm is of a polynomial computational complexity.

**An Illustration Example**

Figure 1 provides a sample network, in which node \(o\) and node \(s\) represent origin and destination, respectively. Node \(v\) is the current node where the traveler needs to decide his/her optimal next action; the data in the parentheses on each arc represents the discrete arc travel time distribution. More precisely, the data within the square brackets indicates a travel time state (a closed interval with specific lower and upper bounds on travel times), and the value beside each travel time state represents the probability of that travel time state occurring at a given time stamp. For example, \(\{(5, 6), 0.95\}\) indicates that there are two possible travel time states, \([5, 6]\) and \([9, 15]\), for the arc from node \(n\) to node \(1\). Further, these two travel time states will occur with probabilities of 0.95 and 0.05, respectively, at the specific time stamp. Using Figure 1 as an example network, we illustrate the method to calculate the label of a node, and further demonstrate the process to identify the optimal next arc under Routing Policy II. Note that the method/process also works for Routing Policy I and Policy K. Without loss of generality, we assume that the traveler arrives at node \(n\) at the current time. According to Figure 1, he/she has two options, Arc \((v, 1)\), and Arc \((v, 2)\), to further continue the trip. The available real-time information at this time is also shown in Figure 1. The process introduced in Section 3.2 to label nodes in a reverse order (such as node \(s\), node \(3\), node \(2\), and node \(1\)) is demonstrated through Equations **Error! Reference source not found.** - (12).

We illustrate the process in detail hereafter. First, the expected minimum travel time variance at the destination node can be set to zero in Equation **Error! Reference source not found..** Then, we go backward to check node 3 (or node 2) since \(\delta^+(s) = \{(3, s), (2, s)\}\). To do this, we have three important observations: (i) There is only one arc emanating from node 3 to node \(s\) (i.e. \(\delta^+(3) = \{(3, s)\}\)). Therefore, the expected minimum travel time variance at node 3 only depends on the travel time states on Arc \((3, s)\); (ii) With the given short-term distribution of Arc \((3, s)\) in Figure 1, we know that Arc \((3, s)\) has three possible states, and each manifests with a certain probability; and (iii) At a single instant, there is only one traffic state appearing on Arc \((3, s)\), such as \([3, 5], 0.6\), which results in the minimum travel time variance at this instant. Based on this idea, we obtain the following results for node 3. There are three possible minimum travel time variances \((0.33, 0.75, 2.08)\)\(^1\) for Arc \((3, s)\) with the corresponding probabilities \((0.6, 0.2, 0.2)\). The expected minimum travel time variance on Arc \((3, s)\) is given by Equation **Error! Reference source not found..**

\[
\text{var}(c_{ss}(t)) = 0
\]  

\(^1\)Note that the travel time value in each individual travel time state (an interval) follows a uniform distribution. Therefore, the travel time variance of each travel time state on a arc can be calculated based on the variance expression for the uniform distribution, such as \(0.33 = (5 - 3)^2 / 12\) for the first travel time state on arc \((3, s)\).
Equation 10
\[
\text{var}(c_{3s}(t)) = E\left[\min\{\text{var}(c_{3s}(t)) + \text{var}(c_{ss}(t))\}\right] \\
= E\left[\min\{\text{var}(c_{3s}(t))\}\right] \\
= E\left[\min\{0.33,0.6,0.75,0.2,2.08,0.2\}\right] \\
= 0.33*0.6+0.75*0.2+2.08*0.2 = 0.76
\]
\[
\text{var}(c_{2s}(t)) = E\left[\min\{\text{var}(c_{2s}(t)) + \text{var}(c_{ss}(t))\}\right] \\
= E\left[\min\{\text{var}(c_{2s}(t))\}\right] \\
= E\left[\min\{0.33,0.4,0.083,0.6\}\right] \\
= 0.33*0.4+0.083*0.6=0.18
\]

**FIGURE 1** An illustration example

Node 2 is similar to Node 3 since only one arc is incident from it. The associated variance is directly given in Equation Error! Reference source not found. Similarly, the algorithm moves backward and calculates the label for node 1. The situation of Node 1 entails some more discussion. Figure 1 shows that there are two arcs emanating from node 1 (that is, \(\delta^{-}(1) = \{(1,2),(1,3)\}\)) and each arc in \(\delta^{-}(1)\) has multiple possible travel time states. It indicates that there are two sets of paths going through node 1 to the destination and each set of paths may have multiple possible travel times. Correspondingly, \(\text{var}(c_{1s}(t))\), the expected minimum travel time variance from node 1 to the destination, can be expressed as \(E\left[\min\{\text{var}(c_{13}(t)) + \text{var}(c_{3s}(t)), \text{var}(c_{12}(t)) + \text{var}(c_{2s}(t))\}\right]\) which is the first step in Equation (12). With the known values of \(\text{var}(c_{3s}(t))\) and \(\text{var}(c_{2s}(t))\) given by Equation Error! Reference source not found., and Equation Error! Reference source not found., as well as the known values of \(\text{var}(c_{13}(t))\) and \(\text{var}(c_{12}(t))\) given by the short-term travel time distribution, we specify the expression of \(\text{var}(c_{1s}(t))\) in the second step of Equation (12).

\[
\text{var}(c_{1s}(t)) = E\left[\min\{\text{var}(c_{13}(t)) + \text{var}(c_{3s}(t)), \text{var}(c_{12}(t)) + \text{var}(c_{2s}(t))\}\right] \\
= E\left[\min\{0.85,0.8,1.51,0.15,2.85,0.05,0.52,0.1,2.27,0.7,2.27,0.2\}\right] \\
= 0.52*0.1+0.85*0.8*0.9+1.51*0.15*0.9+2.27*(0.7+0.2)*0.05=0.97
\]
To illustrate the procedure to determine $\text{var}(c_{15}(t))$, we apply the proposed sub-algorithm in the Routing Algorithm section. At a given time instance, $\text{var}(c_{15}(t))$ depends on the expected travel time variances of the candidate path sets which pass through either Arc $(1,2)$ or Arc $(1,3)$, and each candidate path presents a possible travel time variance. For example, the travel time variance of the path including Arc $(1,3)$ can be 0.8 with the probability 0.85, which is an element in the set of $\{0.85,0.8;1.51,0.15;2.85,0.05\}_{13}$. Similarly, the travel time variance of the path going through Arc $(1,2)$ can be one element, such as $\{0.52,0.1\}_{12}$ in $\{0.52,0.1;2.27,0.9\}_{12}$. Accordingly, the minimum travel time variance of the path starting from node 1 to the destination at this time instance is the smaller one; that is, it is 0.52 with the probability $1.20 \times 0.72$. Along with this, by enumerating all the combinations of the possible travel time variances between the candidate paths, the algorithm is able to specify all possible minimum travel time variances as well as their probabilities, and determine the value of $\text{var}(c_{15}(t))$.

To further articulate the steps, the algorithm uses $\varphi$ in Equation (13) to denote all possible travel time variances of the candidate path sets going through the arcs in $\delta^-(1)$, and adopts $\varphi'$ in Equation (14) to represent the sorted set of $\varphi$. Note that each element in $\varphi'$ and $\varphi$ are indexed by the candidate arc in $\delta^-(1)$. It can be observed that $\{2.85,0.05\}_{13}$ in $\varphi'$ will never be the minimum travel time variance of the paths from node 1 since the travel time variance 2.85 is greater than all corresponding values in $\{0.52,0.1;2.27,0.9\}_{12}$. Hence, the possible minimum values in $\varphi'$ shrink to the set of $\Gamma$ in Equation (15). Each value in $\Gamma$ corresponds to a combination of the travel time variances on all candidate paths. For example $\{1.51\}_{13}$ corresponds to the candidate path passing through Arc $(1,3)$ and has a variance 1.51 with probability 0.15. The candidate path passing through Arc $(1,2)$ has a variance 2.27 with probability 0.9. Consequently, $\text{var}(c_{15}(t))$ is calculated by the third step in Equation (12).

\[
\varphi = \{\{0.85,0.8\}_{13},\{1.51,0.15\}_{13},\{2.85,0.05\}_{13},\{0.52,0.1\}_{12},\{2.27,0.7\}_{12},\{2.27,0.2\}_{12}\} \tag{13}
\]

\[
\varphi' = \{\{0.52,0.1\}_{12}, \{0.85,0.8\}_{13}, \{1.51,0.15\}_{13}, \{2.27,0.9\}_{12}, \{2.85,0.05\}_{13}\} \tag{14}
\]

\[
\Gamma = \min(\varphi') = \left\{\{0.52,0.1\}_{12},\{0.85,0.8;1.51,0.15;2.85,0.05\}_{13},\{0.85,0.8\}_{13},\{2.27,0.9\}_{12},\{1.51,0.15\}_{13},\{2.27,0.9\}_{12},\{2.85,0.05\}_{13}\right\} \tag{15}
\]

With the given real-time information on Arc $[n,1]$ and Arc $[n,2]$, the algorithm is able to identify the optimal next Arc $(n,2)$ for the given trip under the Routing Policy II (see Equation (5)). The corresponding calculation is demonstrated in Equation (16).

\[
\alpha^*(t) = \text{argmin}\{E(\text{var}[c_{n1}(t)]) + \text{var}(c_{15}(t)); E(\text{var}[c_{n2}(t)]) + \text{var}(c_{25}(t))\} = \text{argmin}\{E(\text{var}[c_{n1}(t)]) + 0.97; E(\text{var}[c_{n2}(t)]) + 0.18\} = \text{argmin}\{(6-5)^2/12 \times 0.95 + (15-9)^2/12 \times 0.05 + 0.97; (5-2)^2/12 \times 0.5 + (10-8)^2/12 \times 0.5 + 0.18\} = \text{argmin}\{1.20;0.72\} = (v,2) \tag{16}
\]

**NUMERICAL EXPERIMENTS**

The experiments in this section first focus on the proposed routing framework composed of the designed online routing policies/algorithm, information fusion model and test network. Based upon the framework and the information fusion model, this study demonstrates the efficiency and applicability of the proposed routing algorithms, and further test the impact of the information quality on online routing.
**Study Network**

The experiments are conducted through traffic simulation on the Borman Expressway network, which consists of a 16 mile section of I-94 (called the Borman Expressway), I-90 toll freeway, I-65, and the surrounding arterials. The Borman Expressway network is a highly congested freeway with 30 to 60 percent semi-trailer truck traffic. To manage traffic under incidents and during peak period congestion, an advanced traffic management system has been installed on the Borman network to provide travelers real-time traffic information. With a given O-D, the first test network consists of 29 nodes, and 46 arcs whose boundary is demarcated by the dotted line in Figure 2. Other sub-networks with a larger size are also used in the experiments to test computational efficiency.

**Synthetic Real-Time Traffic Information**

To analyze the proposed online stochastic routing algorithm, it is necessary to generate real-time traffic information containing travel time ranges, time delays, and information accuracy (1) so that the short-term arc travel time distribution is accessible to the traveler. Due to the non-availability of field traffic flow data, this study develops a procedure to generate synthetic real-time travel time information using the DYNASMART simulator.

![Study network](image)

**FIGURE 2 Study network**

The designed procedure first applies the simulation data to develop the long-term historical arc travel time distribution (a histogram), which is used as the starting point for the applied information fusion model (1). This process is briefly described as following. Based on the arc travel time data collected during the first 50 days of the simulation experiments, the arc travel times fall in different travel time states by dividing the range of travel time data (the interval between the maximum and minimum travel time) using a fixed interval length. The probability of each state is estimated as the ratio of the travel time data falling in the corresponding travel time slot to all travel time data obtained during the experiments. Travel time states vary across individual arcs.

Next, real-time information is produced by combining the long-term historical arc travel time distribution and the ground/field truth traffic condition which is represented by the traffic on a certain day in the simulation. More specifically, we first randomly generate the time delay ($\Delta t_a(t)$) and the information accuracy ($p^g_a(t)$), which represents the information quality of real-time travel time
information at time $t$. The time delay, $\Delta t_a(t)$ means the corresponding field truth travel time is $c_a(t - \Delta t_a)$. Furthermore, the real-time travel time interval $[l_a^u(t), u_a^u(t)]$ is determined using a set of steps based on replicating two random scenarios that can occur in the real world: (i) $[l_a^u(t), u_a^u(t)]$ contains the actual travel time value, and (ii) $[l_a^u(t), u_a^u(t)]$ does not contain the actual travel time value. More specifically, a random value $\nu$ is used to simulate these two scenarios. If $\nu < p_a^r$, a real-time travel time interval $[l_a^u(t), u_a^u(t)]$ covering the field truth travel time value $c_a(t - \Delta t_a)$ is generated using the travel time states in the previous short-term distribution as a benchmark for the collected raw field data. If $\nu > p_a^r$, a real-time travel time interval $[l_a^u(t), u_a^u(t)]$ outside the field truth travel time value $c_a(t - \Delta t_a)$ is generated.

**Experiments**

Using synthetic traffic data, this study conducts numerical experiments for 30 days. For each day, the simulation is executed for 90 minutes, using the first 30 minutes as the warm-up time period. Traffic variability is introduced by randomly fluctuating the O-D demands within a given range. The procedure of the experiments on each day includes two main parallel components. First, from the simulation time stamp 30 minutes to 70 minutes, real-time travel time information is randomly generated for each arc after every two minutes. Accordingly, the short-term arc travel time distribution is updated through the information fusion model every two minutes. In this context, the initial short-term arc travel time distribution at the beginning of a day is the long-term historical travel time distribution for that arc. Second, for a given O-D pair, a traveler is routed by the five routing guidelines differentiated by the routing algorithm and the applied travel time information. Namely, at each intersection the optimal next arc is determined by one of four routing guidelines: (i) $\Sigma^{fl}$: following the proposed Routing Policy I and incorporating the short-term arc travel time distribution; (ii) $\Sigma^{H}$: following the proposed Routing Policy I but incorporating the long-term historical arc travel time distribution; (iii) $\Sigma^{fl}$: following the proposed Routing Policy II and incorporating short-term arc travel time distribution; (iv) $\Sigma^{K}$: following the proposed Routing Policy K and incorporating the short-term arc travel time distribution; or (v) $\Sigma^{r}$: following the pre-defined routing path identified by the real-time network traffic condition when the traveler was at the origin. $\Sigma^{*}$ is used to represent the best possible path on the ground. It can be identified by the ground traffic data available in the simulation. The traveler may experience different paths based on these four routing guidelines. During the procedure, we collect the following data at each intersection: the computational time to find the optimal next arc, the expected travel time as well as the variance from the current nodes to the destination; and the experienced path.

The five proposed routing guidelines seek to demonstrate the performance of the proposed online routing policies in the following aspects: (i) The effect of real-time information on the proposed routing policy is explored by comparing the shortest paths resulting from guidance $\Sigma^{fl}$ and $\Sigma^{H}$; (ii) The efficiency of Routing Policy I is explored by comparing the shortest path resulting from $\Sigma^{fl}$ with the predefined shortest path resulting from $\Sigma^{r}$ as well as $\Sigma^{*}$; (iii) The efficiency of Routing Policy II is investigated by comparing the variances of the resulting paths en route when $\Sigma^{fl}$, $\Sigma^{fl}$ or $\Sigma^{H}$ are employed to guide the traveler through the network, respectively. In addition, the comparison of the path travel time resulting from $\Sigma^{fl}$ to $\Sigma^{*}$ indicates the effect of minimizing path travel time variance on the resulting path travel time, and (iv) The effects of the traveler’s preferences related to the travel time and travel time variance are demonstrated through the outcome of $\Sigma^{K}$. The experiment results discussed in the next section address these aspects.
Experimental results

Efficiency of Policy I

The experiment results associated with Routing Policy I are shown in Table 1, where the first and second columns are the routing guidelines and the information accuracy, respectively; the third and fourth columns provide the corresponding average path travel time, and the probability that the traveler experiences the best possible path on the ground over the 30-day experiment period. By analyzing the results in Table 1, we obtain the following observations and insights. First, \( \Sigma^{fl} \), following Routing Policy I, is more efficient to find a shorter path in terms of travel time than \( \Sigma^{r} \), following the pre-defined routing, under different information accuracy situations. This point is illustrated by comparing the average path travel time resulting from \( \Sigma^{fl} \) to the corresponding values resulting from \( \Sigma^{r} \) under different information accuracies, such as \( p^{r} = 0.9, 0.7 \) or \( 0.5 \), in Table 1. It is found that the former is always smaller than the latter in the experiments. Second, under Routing Policy I, \( \Sigma^{fl} \), employing the short-term arc travel time distribution, always leads to a path with a smaller average travel time than \( \Sigma^{H} \), applying long-term historical arc travel time distribution. Moreover, higher information accuracy results in shorter path. These observations imply that the shortest path resulting from Routing Policy I benefits from the real-time travel time information and the embedded data fusion techniques. Third, Routing Policy I significantly improves the opportunity to find the possible best path on the ground when the real-time information is provided with high information accuracy. For example, Table 1 shows that when the real-time travel time information accuracy, \( p^{r} \) is equal to 0.9, the probability that the path resulting from \( \Sigma^{fl} \) equals to \( \Sigma^{*} \) (the possible best path) on the ground is 0.73, which is much higher than all other values in column A. All of these observations indicate that the proposed Routing Policy I combined with short-time arc travel time distribution is an efficient routing guideline under dynamic traffic condition.

<table>
<thead>
<tr>
<th>Guidance</th>
<th>Information accuracy</th>
<th>Average Travel Time (minutes)</th>
<th>A</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \Sigma^{fl} )</td>
<td>( p^{r}=0.9 )</td>
<td>13.32</td>
<td>0.73</td>
</tr>
<tr>
<td>( \Sigma^{fl} )</td>
<td>( p^{r}=0.7 )</td>
<td>14.74</td>
<td>0.33</td>
</tr>
<tr>
<td>( \Sigma^{fl} )</td>
<td>( p^{r}=0.5 )</td>
<td>15.42</td>
<td>0.13</td>
</tr>
<tr>
<td>( \Sigma^{r} )</td>
<td>( p^{r}=0.9 )</td>
<td>14.17</td>
<td>0.47</td>
</tr>
<tr>
<td>( \Sigma^{r} )</td>
<td>( p^{r}=0.7 )</td>
<td>14.93</td>
<td>0.27</td>
</tr>
<tr>
<td>( \Sigma^{r} )</td>
<td>( p^{r}=0.5 )</td>
<td>15.51</td>
<td>0.20</td>
</tr>
<tr>
<td>( \Sigma^{H} )</td>
<td>-</td>
<td>15.91</td>
<td>0.03</td>
</tr>
<tr>
<td>( \Sigma^{*} )</td>
<td>-</td>
<td>12.75</td>
<td>-</td>
</tr>
</tbody>
</table>

A= the probability to have the possible best path in terms of travel time on the ground

Efficiency of Policy II

Next, the performance of Routing Policy II, focusing on minimizing the travel time variation en-route, is analyzed. To do so, the experiments compare the predicted travel time variance of the candidate paths en-route, given that the immediate next arc of each candidate path is selected by the Routing Policy I or...
II. The results in Figure 3 indicate that the paths resulting from the guidelines of $\Sigma^H$, $\Sigma^{fI}$, and $\Sigma^{fII}$ (following Routing Policy I and II, respectively, but integrating the short-term arc travel time distribution) have less variance than the path from $\Sigma^H$ (following Routing Policy I but using long-term historical arc travel time distribution). Further, the path from Routing Policy II has less variance than the path from Routing Policy I. Hence, the proposed Routing Policy II can efficiently find the most reliable path en-route; in addition it benefits from the embedded information fusion model.

![Graph showing predicted travel time variance en-route](image)

**FIGURE 3** The predicted travel time variance en-route under the guideline of $\Sigma^H$, $\Sigma^{fI}$, or $\Sigma^{fII}$

*Routing Policy K*

The study also evaluates the performance of Routing Policy K, assigning different weights to travel time and travel time variance. The associated results demonstrate that as a smaller weight is assigned to travel time, Routing Policy K will lead to a path with a relatively large travel time (see the values in third column of Table 2), but with a relatively small variance (see the curves in Figure 4), and vice versa. Accordingly, it will have less chance to end with the best ground path in terms of travel time (see the values in fourth column of Table 2). These observations reflect the trade-off between travel time and travel time variance in the decision-making process of online routing under dynamic traffic conditions. Moreover, Figure 4 shows that as the trip gets closer to the destination, the difference between the paths resulting from Routing Policies I and II becomes negligible. This is consistent to the reality that traffic variation will become relatively smaller as the path becomes shorter. Correspondingly, travel time will dominate the solution of the routing policy/algorithm in our approach, though travel time variance is given a relatively high weight. The experiments demonstrate that Routing Policy K may generate a customized “best” path since it integrates the traveler preferences related to travel time and travel time variance in the route choice decision-making process.
FIGURE 4 The predicted travel time variance at each decision intersection guided by $\Sigma^k$ given that the weights of travel time are assigned as 1.0, 0.8, 0.6, 0.4, 0.2, and 0

Table 2 Average routing results over 30 days following guidance $\Sigma^k$

<table>
<thead>
<tr>
<th>Guidelines</th>
<th>Weight of travel time</th>
<th>Average Travel Time (minutes)</th>
<th>$A$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\Sigma^k$</td>
<td>1.0</td>
<td>13.32</td>
<td>0.73</td>
</tr>
<tr>
<td></td>
<td>0.8</td>
<td>13.36</td>
<td>0.73</td>
</tr>
<tr>
<td></td>
<td>0.6</td>
<td>13.44</td>
<td>0.67</td>
</tr>
<tr>
<td></td>
<td>0.4</td>
<td>13.36</td>
<td>0.70</td>
</tr>
<tr>
<td></td>
<td>0.2</td>
<td>14.11</td>
<td>0.37</td>
</tr>
<tr>
<td></td>
<td>0.0</td>
<td>15.43</td>
<td>0.13</td>
</tr>
<tr>
<td>$\Sigma^H$</td>
<td>1.0</td>
<td>15.91</td>
<td>0.03</td>
</tr>
<tr>
<td>$\Sigma^*$</td>
<td>1.0</td>
<td>12.75</td>
<td>1</td>
</tr>
</tbody>
</table>

$A$ = the probability to have the possible best path in terms of travel time on the ground

**Computational performance**

Using Policy I as an example, the experiments analyze the computational times to find the optimal next arcs in networks of different sizes. The results in Table 3 demonstrate that using a computer with configuration: 4.0GB RAM, Intel(R) Core(TM)i3 (64-bit) M330@2.13GHz processor, it takes 0.74 seconds execution time to find the optimal next arc in a network with 203 arcs. As the network size become smaller, the needed computational time reduces. These test results illustrate the applicability of the proposed approach in the following two facets. First, for a very large network which has much more than 200 arcs, the arcs far away from the current decision node are not of significant relevance to determine the current optimal next arc since a relatively long travel time is required to approach those arcs and the traffic conditions have already changed dynamically. Second, the lower computational time ensures that the proposed algorithm finds the optimal next arc before the availability of the new information (usually real-time information is generated with a time interval more than 30 seconds, which is much longer than the maximum running time, 0.74 seconds, in our experiments). Overall, the results indicate that Routing Policies I and II perform efficiently to provide travelers optimal online routing in terms of the travel time or travel time variance under dynamic traffic conditions.
Table 3 The computational load

<table>
<thead>
<tr>
<th>Experiments</th>
<th># of Arcs</th>
<th>Computational Time (Seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>41</td>
<td>0.16</td>
</tr>
<tr>
<td>2</td>
<td>74</td>
<td>0.39</td>
</tr>
<tr>
<td>3</td>
<td>100</td>
<td>0.47</td>
</tr>
<tr>
<td>4</td>
<td>123</td>
<td>0.71</td>
</tr>
<tr>
<td>5</td>
<td>203</td>
<td>0.74</td>
</tr>
</tbody>
</table>

**CONCLUSION AND FUTURE WORK**

The proposed work arises in the context of the effective deployment of ITS, which enables real-time traffic conditions to be tracked by various advanced traffic sensors. Thereby, the integration of the advanced traffic detectors, with well-developed data fusion and computer techniques, enables stochastic real-time travel time information to be available to travelers, and impacts their path choices en-route. In this context, travelers need reliable online stochastic route guidance, which can take into account both the quality of the real-time traffic information as well as the variability in the dynamic traffic conditions. Hence, this study is motivated by the need to provide reliable routing strategies. It is additionally motivated by the notion of analyzing the optimal paths at decision nodes, which arises in the context of vehicle-to-vehicle communications based traffic systems where real-time traffic information propagates among individual vehicles equipped with wireless communications capabilities, and where vehicles evaluate their routes at discrete locations based on the seamless spatio-temporal exchange of information with other vehicles.

This study develops online routing policies/algorithms where the real-time arc travel time is integrated with the discrete short-term arc travel time distribution. The proposed Routing Policy I focuses on finding the shortest path based on the expected minimum travel time from the current decision node to the destination, while Routing Policy II puts more emphasis on minimizing the travel time variance of a path from the current decision node/intersection to the destination. Based on combining Routing Policies I and II, we also propose Routing Policy K, which considers the trade-offs between travel time and travel time variance in the route choice decision-making process. A modified label-correcting algorithm is designed to solve for the shortest path resulting from the proposed routing policies, in polynomial running time bounded by $O(mn^2)$.

Numerical experiments provide the following insights. First, by using the short-term arc travel time distributions, the proposed online routing polices generate better paths than the those that rely on the long-term historical arc travel time distributions. Thereby, higher information accuracy leads to better paths (closer to the best path on the ground). Second, under the same information accuracy, the proposed online routing policies, which adaptively incorporate the online traffic information, usually lead to better paths than paths pre-determined at the origin. Third, Routing Policy II results in a path with a smaller variance than Routing Policy I. Fourth, Routing Policy K can generate a customized “best” path by accounting for traveler preferences in terms of travel time and travel time variance.
Finally, the proposed routing algorithm is computationally efficient, and is hence amenable to online applications.

A potential future effort stemming from this study is to integrate a more comprehensive travel time reliability measure into Policy II, and further develop a systematic method to determine the weights associated with the traveler’s preferences in terms of travel time and travel time reliability in Routing Policy K. Another planned effort is to develop an efficient algorithm which identifies the dynamic local network associated with the current origin and destination at each time stamp; it can potentially further reduce the computational load at each decision node.

REFERENCES


