Abstract—As travel times in road networks are dynamic and uncertain, it is difficult and time-consuming to search for the least expected time path in large-scale networks. This paper addresses the problem of finding the least expected time path in stochastic time-dependent (STD) road networks. A stochastic travel speed model is proposed to represent STD link travel times. It is proved that the link travel times in STD networks satisfy the stochastic first-in-first-out (S-FIFO) property. Based on this S-FIFO property, an efficient multicriteria A∗ algorithm is proposed to exactly determine the least expected time path in STD networks. Computational results using several large-scale road networks show that the proposed algorithm has a significant computational advantage over existing solution algorithms without the S-FIFO property.

Index Terms—Least expected time path, shortest path problem, stochastic first-in-first-out (S-FIFO), stochastic time-dependent (STD) network.

NOMENCLATURE

\(E(\cdot)\) Mean of a random variable.

\(\text{Var}(\cdot)\) Variance of a random variable.

\(\text{Cov}(\cdot, \cdot)\) Covariance between two random variables.

\(\Phi_X(\cdot)\) Cumulative distribution function (CDF) of random variable \(X\).

\(\Phi^{-1}_X(\cdot)\) Inverse CDF of random variable \(X\).

\(A\) A set of links.

\(a_{ij}\) A link connecting node \(i\) and node \(j\).

\(D_{y_i}y_{y_i}\) Travel distance distribution from time instances \(y_i\) and \(y_{y_i}\).

\(G\) A stochastic time-dependent (STD) network.

\(N\) A set of nodes.

\(p^r_s\) A path between origin \(r\) and destination \(s\).

\(P^r_i\) A set of paths from origin \(r\) to node \(i\).

\(T^r_s(y_r)\) Path travel time distribution.

\(T_{ij}(y_i)\) Link travel time distribution.

\(V^m_{ij}\) Travel speed distribution of link \(a_{ij}\) during time interval \((\Delta_{n-1}, \Delta_n)\).

\(Y_i\) Stochastic arrival time at node \(i\).

\(y_i\) Deterministic arrival time at node \(i\).

\(y_r\) Departure time at origin \(r\).

\(z_{r}\) Inverse CDF of standard normal distribution at \(\lambda\) confidence level.

\(\lambda\) Confidence level \(\lambda \in (0, 1)\).

\(\Delta\) Length of discrete time interval.

\(\Omega\) Period of interest.

I. INTRODUCTION

WITH ADVANCES in information technology, much attention has been given to the development of advanced traveler information systems for enhancing the effectiveness of network demand and supply management [1]–[5]. It has been well recognized that advanced traveler information systems not only can help travelers to make better route choice decisions but also can improve overall network traffic conditions. The shortest path algorithm is at the core of advanced traveler information systems, in which the shortest path between origin and destination nodes should be found in a quick and accurate manner.

Early studies addressed time-dependent shortest path problems based on a deterministic assumption of link travel times [6]–[8]. Efficient solution algorithms have been developed for solving such shortest path problems when link travel times are first-in-first-out (FIFO) consistent [9]. FIFO implies that vehicles entering a link earlier cannot arrive at the end of the link later. In other words, overtaking is not allowed in the FIFO link. Although overtaking behavior is common in...
real road networks, this FIFO property seems valid in the context of routing applications because vehicles are assumed to be traveling at the same average speed [7]. Accordingly, Sung et al. [7] found that link travel times in road networks are FIFO consistent. Qian et al. [10] pointed out that link travel times in signalized road networks are also FIFO consistent.

Link travel times in congested urban road networks, however, are highly stochastic due to traffic demand fluctuations and capacity degradations [11]–[13]. The link travel times should therefore be represented as STD variables, the distributions of which vary with the time of day. The resulting STD network will provide a more appropriate representation of actual traffic conditions than the traditional deterministic network [14]. For providing credible route guidance, it is therefore necessary to investigate shortest path problems in STD networks.

In the literature, link travel times in STD networks are commonly formulated using stochastic link travel time model (S-LTM) [14]–[16]. In the S-LTM, the link travel time distribution experienced by a vehicle depends on the time instance when the vehicle enters the link, and the travel speed of the vehicle is assumed to be fixed while the vehicle is traveling on the link. Using the S-LTM, many researchers have found that link travel times in STD networks do not satisfy stochastic first-in-first-out (S-FIFO) property [14]–[16]. S-FIFO is a stochastic extension of the traditional FIFO property, which implies that at any probability level, a vehicle entering the link earlier cannot arrive at the end of the link later.

Because of this violation of the S-FIFO property, few efficient solution algorithms have been developed for finding the optimal path in STD networks. Miller-Hooks and Mahmassani [14] proposed a multicriteria label-correcting algorithm to determine the least expected time path by generating all nondominated paths in the network for all possible departure times. To make the algorithm tractable, the discretization of link travel times and departure times is required. This label-correcting algorithm can exactly determine the least expected time path. However, the generation of nondominated paths for all possible departure times is computationally intensive for large-scale road networks. Fu and Rilett [15] presented a heuristic solution algorithm based on the K shortest path algorithm for finding the least expected time path. The proposed heuristic algorithm, nevertheless, may miss the optimal path completely.

This paper aims to reinvestigate the S-FIFO property of link travel times and to devise efficient solution algorithms for solving shortest path problems in STD networks. This paper extends the previous work in following aspects.

First, it is found in this paper that the link travel time distribution satisfies the S-FIFO property. The violation of the S-FIFO property in the S-LTM is due to the fixed travel speed assumption. By relaxing this assumption, we propose a generalized model, i.e., the stochastic travel speed model (S-TSM), in which travel speeds of a vehicle can be varied while the vehicle is traveling on the link. Using the S-TSM, the S-FIFO property of link travel time distribution is rigorously proved.

Second, an efficient solution algorithm is proposed in this paper for solving the least expected time path problem. Based on the S-FIFO property, the least expected time path problem can be formulated as a class of multicriteria shortest path problems and solved by determining a set of nondominated paths under first-order stochastic dominance (FSD) [17]. An efficient multicriteria A∗ algorithm is developed for finding the least expected time path without generating all FSD nondominated paths. Computational results show that the proposed algorithm has a significant computational advantage over the existing algorithm without the S-FIFO property [14].

Finally, a case study is carried out using data from a real-world advanced traveler information system in Hong Kong. Numerical results of the case study demonstrate that the proposed algorithm can determine the least expected time path in large-scale road networks within a reasonable computing time. The proposed solution algorithm, therefore, is potentially useful for providing real-time route guidance in large-scale networks with STD link travel times.

The rest of this paper is organized as follows. Section II formulates the least expected time path finding problem in STD networks. Section III describes the proposed S-TSM and discusses the S-FIFO property of link travel times. Section IV presents the solution algorithm for finding the least expected time path in STD networks. Section V reports the case studies conducted on several real road networks. Finally, conclusions are presented together with recommendations for further study.

II. PROBLEM FORMULATION

Let $G=(N, A, \Omega)$ be a directed STD network, where $N$ is the set of nodes, $A$ is the set of links, and $\Omega$ is the period of interest within a day. Along the line of pervious work [14]–[16], the period of interest $\Omega$ is described as a set of discrete time intervals $(\ldots, \Delta_n = n \Delta, \ldots)$, where $n$ is an integer, and $\Delta$ is the length of a discrete time interval. Each node $i$ has a set of successor nodes $SCS(i) = \{j : a_{ij} \in A\}$ and a set of predecessor nodes $PDS(i) = \{k : a_{ki} \in A\}$. Each link $a_{ij} \in A$ has a tail node $i \in N$, a head node $j \in N$, and a set of random travel times varying over the period of interest. $T_{ij}(y_i)$ denotes a random travel time for vehicles entering link $a_{ij}$ at time instance $y_i \in (\Delta_{n-1}, \Delta_n)$. For notational consistency, capital letters represent random variables, lowercase letters represent deterministic variables, and bold letters represent sets of symbols throughout this paper.

Suppose that $r \in N$ and $s \in N$ represent the origin and destination nodes, respectively. Let $p^{rs}$ be a path from origin $r$ to destination $s$, $T^{rs}(y_r)$ denotes the travel time for a journey starting at a given departure time $y_r$ and going through path $p^{rs}$. In congested urban road networks, waiting at network intersections, generally, is not allowed [14], [15]. Based on this assumption, $T^{rs}(y_r)$ can be calculated as

$$T^{rs}(y_r) = \sum_{a_{ij} \in A} T_{ij}(Y_i) \delta_{ij}^{rs} \quad (1)$$

where $Y_i$ is the arrival time at node $i$, and $\delta_{ij}^{rs}$ is the path–link incidence variable; $\delta_{ij}^{rs} = 1$ means that link $a_{ij}$ is on path $p^{rs}$ and $\delta_{ij}^{rs} = 0$ otherwise. Generating path travel time in STD networks is not trivial since link travel time $T_{ij}(\cdot)$ depends on...
the time instance a vehicle arrives at node \(i\), but this arrival time \(Y_i\) itself is a random variable.

Let \(E(\cdot)\) and \(\text{Var}(\cdot)\) be the mean and the variance of a random variable, respectively. The problem of finding the least expected time path in a STD network can be defined as the following optimization problem:

\[
\text{Min} \ E(T^{rs}(y_r))
\]

subject to

\[
T^{rs}(y_r) = \sum_{a_{ij} \in A} T_{ij}(Y_i) \delta_{ij}^r
\]

\[
\sum_{j \in SCS(i)} \delta_{ij}^r - \sum_{k \in PDS(i)} \delta_{ki}^r = \begin{cases} 
1 & \forall i = r \\
0 & \forall i \neq r; \ i \neq s \\
-1 & \forall i = s
\end{cases}
\]

\[
\delta_{ij}^r \in \{0, 1\}, \ \forall a_{ij} \in A.
\]

Equation (2) is the expected path travel time \(E(T^{rs}(y_r))\), which travelers want to minimize. Equation (3) defines the path travel time distribution. Equation (4) ensures that links on the least expected time path are feasible. Equation (5) is concerned with the link–path incidence variables, which should be binary in nature.

III. REPRESENTATION OF STD LINK TRAVEL TIMES

This section describes the S-TSM for representing STD link travel times. The mathematical formulation of the S-TSM is described in Section III-A. A discrete approximation method to generate the link travel time distribution under the S-TSM is presented in Section III-B.

A. S-TSM

In this paper, the traffic conditions in a congested road network are represented by travel speeds. The travel speeds are direct measurements of many traffic detectors (e.g., loop detectors) and are commonly used by advance traveler information systems for disseminating traffic conditions to road users [18]–[20]. Due to the stochastic nature of congested road networks, link travel speeds are modeled as normal distributions [13], [21]. Let \(V^{rs}_{ij}\) be the travel speed distribution of link \(a_{ij}\) during time interval \((\Delta_{n-1}, \Delta_n)\).

In the S-TSM, the travel time distribution experienced by a vehicle depends on the time instance at which the vehicle enters the link, and travel speed distributions of the vehicle can be varied while the vehicle is traveling on the link. Suppose that a vehicle enters link \(a_{ij}\) at time instance \(y_i\in (\Delta_{n-1}, \Delta_n)\). Using the S-TSM, the travel distance (denoted by \(D^{y_i}_{aij}\)) at time instance \(y_i\in (\Delta_{n-1}, \Delta_n)\) can be calculated as

\[
D^{y_i}_{aij} = V^{rs}_{ij}(\Delta_{n-1} - y_i) + V^{rs}_{ij} + \cdots + V^{rs}_{ij}(\Delta_n - \Delta_{m-1}).
\]

As travel speeds are random variables, so is the travel distance \(D^{y_i}_{aij}\). Let \(d_{ij}\) be the length of link \(a_{ij}\). The probability that the vehicle arrived at head node \(j\) before time instance \(y_a\) can be expressed as

\[
\Pr(D^{y_a}_{aij} \leq d_{ij}) = \lambda.
\]

Let \(\Phi(\cdot)\) and \(\Phi^{-1}(\cdot)\) be the CDF and the inverse CDF of a random variable, respectively. This arrival probability \(\lambda\) can be also expressed in terms of the CDF of the arrival time distribution as

\[
\Phi_{y_a}(y_a) = \lambda
\]

where \(y_a\) is the arrival time distribution of the vehicle arriving at head node \(j\). Based on (8), the inverse CDF of the arrival time distribution at \(\lambda\) confidence level can be calculated as

\[
\Phi^{-1}_{y_a}(\lambda) = y_a.
\]

If \(\Phi^{-1}_{y_a}(\lambda)\) for any confidence level \(\lambda\) can be calculated by (6)–(9), the whole arrival time distribution \(Y_a(y_a)\) can be generated. The method for generating the arrival time distribution is described in Section III-B. Consequently, the link travel time \(T_{ij}(y_i)\), which is required for the vehicle to travel on link \(a_{ij}\), can be expressed as

\[
T_{ij}(y_i) = Y_y(y_i) - y_i.
\]

It can be proved that link travel time \(T_{ij}(y_i)\) under the S-TSM satisfies the S-FIFO property as in the succeeding discussion.

Definition 1: Link travel time distribution \(T_{ij}(y_i)\) is S-FIFO consistent if \(\Phi^{-1}_{T_{ij}(y_i)}(\lambda)\) satisfies \(y_1 < y_2 \Rightarrow \Phi^{-1}_{T_{ij}(y_1)}(\lambda) = y_1 + \Phi^{-1}_{T_{ij}(y_2)}(\lambda) < \Phi^{-1}_{T_{ij}(y_1)}(\lambda) = y_2 + \Phi^{-1}_{T_{ij}(y_2)}(\lambda), \ \forall \lambda \in (0, 1).

Proposition 1: Link travel times under the S-TSM are S-FIFO consistent.

Proof: Assume that two vehicles respectively enter link \(a_{ij}\) at time instances \(y_1\) and \(y_2\) \((y_1 < y_2)\) and that they arrive at head node \(j\) at \(Y_{y_1}(y_i)\) and \(Y_{y_2}(y_i)\), respectively. According to (6)–(9), \(\Phi^{-1}_{T_{ij}(y_1)}(\lambda) < \Phi^{-1}_{T_{ij}(y_2)}(\lambda), \ \forall \lambda \in (0, 1)\), is equivalent to prove \(\Phi^{-1}_{D^{y_1}_{aij}}(\lambda) > \Phi^{-1}_{D^{y_2}_{aij}}(\lambda), \ \forall \lambda \in (0, 1)\), \(\forall y_i \in \Omega\), as follows.

Let \(z_\lambda\) be the inverse CDF of standard normal distribution at \(\lambda\) confidence level.

\[
\Phi^{-1}_{D^{y_1}_{aij}}(\lambda) - \Phi^{-1}_{D^{y_2}_{aij}}(\lambda) = E(D^{y_1}_{aij} + D^{y_2}_{aij}) + z_\lambda \sqrt{\text{Var}(D^{y_1}_{aij} + D^{y_2}_{aij})} - E(D^{y_1}_{aij}) + z_\lambda \sqrt{\text{Var}(D^{y_1}_{aij})} + 2 \text{Cov}(D^{y_1}_{aij}, D^{y_2}_{aij}) + \text{Var}(D^{y_1}_{aij}) - z_\lambda \sqrt{\text{Var}(D^{y_1}_{aij})}
\]

When \(\lambda > 0.5\), \(\Phi^{-1}_{D^{y_1}_{aij}}(\lambda) - \Phi^{-1}_{D^{y_2}_{aij}}(\lambda) > E(D^{y_1}_{aij}) - z_\lambda \sqrt{\text{Var}(D^{y_1}_{aij})} \).

Due to the nonnegative property of link travel distance, we have \(E(D^{y_1}_{aij}) - z_\lambda \sqrt{\text{Var}(D^{y_1}_{aij})} > 0\). Similarly, when \(\lambda < 0.5\), \(\Phi^{-1}_{D^{y_1}_{aij}}(\lambda) - \Phi^{-1}_{D^{y_2}_{aij}}(\lambda) < E(D^{y_2}_{aij}) + z_\lambda \sqrt{\text{Var}(D^{y_2}_{aij})} \).

Therefore, \(\Phi^{-1}_{D^{y_1}_{aij}}(\lambda) > \Phi^{-1}_{D^{y_2}_{aij}}(\lambda), \ \forall \lambda \in (0, 1)\). \square

A simple example is given to illustrate the concept of the S-TSM with a comparison to the traditional S-LTM. Consider a link \(a_{ij}\) with a link length of 600 m. All travel speeds \(V^{rs}_{ij}\) are assumed to follow a normal distribution, and the time interval used is 1 min. The means of the link travel speeds \(E(V^{rs}_{ij})\) are
The standard deviations of the link travel speeds \( \sqrt{\text{Var}(V^n_{ij})} \) are 0.2 \( \times E(V^n_{ij}) \) (coefficients of variation are 0.2). The correlation coefficients between each two travel speed distributions are assumed to be 0.5.

Consider two vehicles entering the link at \( y_1 = 8:00:10 \) and \( y_2 = 8:01:20 \), respectively. Using the S-TSM, arrival time distributions \( Y^n_j(y_1) \) and \( Y^n_j(y_2) \) of the vehicle arriving at head node \( j \) can be calculated and illustrated in Fig. 1(c) (referred to as S-TSM(\( y_1 \)) and S-TSM(\( y_2 \))). The detailed method for generating these two arrival time distributions is given in Section III-B. It is shown in Fig. 1(c) that under the S-TSM, arrival time distributions (and link travel time distributions) are S-FIFO consistent (i.e., \( \Phi^{-1}_{Y^n_j(y_1)}(\lambda) < \Phi^{-1}_{Y^n_j(y_2)}(\lambda), \forall \lambda \in (0, 1) \)). This implies that at any confidence level \( \lambda \), a vehicle entering the link earlier cannot arrive at the end of the link later.

Link travel time distributions can be also calculated under the traditional S-LTM. Let \( \bar{T}^n_{ij}(y_i) \) be the link travel time under the S-LTM experienced by a vehicle entering link \( a_{ij} \) at time instance \( y_i \). According to the definition of the S-LTM, \( \bar{T}^n_{ij}(y_i) \) depends only on travel speed \( V^n_{ij} \) at time instance \( y_i \), and the vehicle’s travel speeds are assumed to be fixed while the vehicle is traveling on the link. Therefore, \( \bar{T}^n_{ij}(y_i) \) can be expressed as
\[
\bar{T}^n_{ij}(y_i) = d_{ij}/V^n_{ij}. \tag{11}
\]

As link travel speed \( V^n_{ij} \) follows a normal distribution, therefore, \( \bar{T}^n_{ij}(y_i) \) is the reciprocal of normal the distribution, which can be approximated as a lognormal distribution [21]. Performing a first-degree Taylor series expansion [21], mean link travel time \( E(\bar{T}^n_{ij}(y_i)) \) can be calculated by dividing the link length \( d_{ij} \) by \( E(V^n_{ij}) \). The standard deviation of the link travel time \( \sqrt{\text{Var}(\bar{T}^n_{ij}(y_i))} \) can be calculated by multiplying the coefficients of variation (i.e., 0.2) by \( E(\bar{T}^n_{ij}(y_i)) \). Fig. 1(b) shows the means of link travel time distributions in this example. Let \( \bar{Y}^n_j(y_i) \) be the arrival time distribution under the S-TLM. It can be easily calculated by \( \bar{Y}^n_j(y_i) = y_i + \bar{T}^n_{ij}(y_i) \).

Fig. 1(c) illustrates arrival time distributions \( \bar{Y}^n_j(y_1) \) and \( \bar{Y}^n_j(y_2) \) at time instances \( y_1 \) and \( y_2 \) (referred to as S-LTM(\( y_1 \)) and S-LTM(\( y_2 \))). It is shown in Fig. 1(c) that the arrival time distributions under the S-LTM violate the S-FIFO property. For example, \( \Phi^{-1}_{\bar{Y}^n_j(y_1)}(0.8) = 8:04:02 < \Phi^{-1}_{\bar{Y}^n_j(y_2)}(0.8) = 8:03:39 \). This indicates that at the same confidence level \( \lambda = 0.8 \), a vehicle entering the link 70 s earlier arrives at the end of the link 23 s later.

It is also shown in Fig. 1(c) that the arrival time distribution \( \bar{Y}^n_j(y_1) \) generated by the S-LTM has a significant bias compared with \( \bar{Y}^n_j(y_1) \) generated by the S-TSM. This bias occurs because the S-LTM uses the fixed travel speed assumption, although traffic conditions are significantly improved while the vehicle is traveling the link.

B. Discrete Method for Generating Link Travel Time Distribution

Because generating the exact continuous distribution of link travel times (i.e., \( T_{ij}(y_i) \)) under the S-TSM has no closed form, a discrete method is proposed here. The CDF of the link travel time distribution is discretized into \( L_1 \) equal-probability intervals \( \lambda = \varepsilon_1, 2\varepsilon_1, \ldots, L_1\varepsilon_1 \), where \( L_1\varepsilon_1 = 1.0 \). The generation of link travel time distribution corresponding to \( L_1 \) discrete points is equivalent to the generation of a sequence of discrete inverse CDFs, i.e., \( \Phi^{-1}_{T_{ij}(\lambda)}(\lambda) = \varepsilon_1, 2\varepsilon_1, \ldots, L_1\varepsilon_1 \).

Given a vehicle starting at tail node \( i \) at time instance \( y_i \in (\Delta_n-1, \Delta_n) \), the travel distance \( D^\Delta_{y_i} \) at time instance \( \Delta_n \) can be expressed as
\[
D^\Delta_{y_i} = V^n_{ij}(\Delta_n - y_i) + V^n_{ij+1} + \cdots + V^n_{ijm}. \tag{12}
\]

The mean and the standard deviation of the travel distance can be calculated as
\[
E(D^\Delta_{y_i}) = E(V^n_{ij}) (\Delta_n - y_i) + \sum_{k=n+1}^{k=m} E(V^n_{ij}) \Delta \tag{13}
\]
\[
\text{Var}(D^\Delta_{y_i}) = \text{Var}(V^n_{ij}) (\Delta_n - y_i)^2 + \sum_{k=n+1}^{k=m} \text{Var}(V^n_{ij}) \Delta^2
\]
\[
+ \sum_{k=n+1}^{k=m} 2\text{Cov}(V^n_{ij}, V^n_{ij+1}) \Delta(\Delta_n - y_i)
\]
\[
+ \sum_{k=n+1}^{k=m} \sum_{w=n+1}^{w=m} 2\text{Cov}(V^n_{ij}, V^n_{ij+1}) \Delta^2 \tag{14}
\]
where \( \text{Cov}(V^n_{ij}, V^n_{ij+1}) \) is the travel speed temporal covariance between two time intervals. This temporal covariance can be also calculated by
\[
\text{Cov}(V^n_{ij}, V^n_{ij+1}) = \rho^{V^n_{ij}}_{ij} \sqrt{\text{Var}(V^n_{ij})} \sqrt{\text{Var}(V^n_{ij+1})} \tag{15}
\]
where \( \rho^{V^n_{ij}}_{ij} \in [-1, +1] \) is the correlation coefficient between \( V^n_{ij} \) and \( V^n_{ij+1} \); \( \rho^{V^n_{ij}}_{ij} = +1 \) and \( \rho^{V^n_{ij}}_{ij} = -1 \) are the perfect positive and negative relationships between these two variables.
Let $\Phi_{D_{y_{i-1}}}(d_{ij}) = 1 - \beta_{m-1}$ and $\Phi_{D_{y_{i}}}(d_{ij}) = 1 - \beta_{m}$ be the CDFs of $D_{y_{i-1}}$ and $D_{y_{i}}$ for link length $d_{ij}$. Then, $\Phi_{T_{ij}}^{-1}(\beta_{m-1})$ and $\Phi_{T_{ij}}^{-1}(\beta_{m})$ can be expressed as

$$\Phi_{T_{ij}}^{-1}(\beta_{m-1}) = \Delta_{m-1} - y_{i}$$

(16)

$$\Phi_{T_{ij}}^{-1}(\beta_{m}) = \Delta_{m} - y_{i}.$$  

(17)

For $\lambda \in (\beta_{m-1}, \beta_{m})$, the inverse CDF of the link travel time distribution satisfies $\Phi_{T_{ij}}^{-1}(\lambda) \in (\Delta_{m-1} - y_{i}, \Delta_{m} - y_{i})$. At time instance $y = \Phi_{T_{ij}}^{-1}(\lambda) + y_{i}$, travel time distribution $D_{y_{i}}^{y}$ can be calculated as

$$D_{y_{i}}^{y} = D_{y_{i}}^{\Delta_{m-1}} + V_{ij}^{m}(y - \Delta_{m-1}).$$

(18)

The inverse CDF of $D_{y_{i}}^{y}$ can be expressed as

$$\Phi_{D_{y_{i}}^{y}}^{-1}(\lambda) = d_{ij}.$$  

(19)

As travel distance $D_{y_{i}}^{y}$ follows a normal distribution, (19) can be formulated as the following quadratic equation:

$$d_{ij} = E(D_{y_{i}}^{y}) + z_{\lambda}\sqrt{\text{Var}(D_{y_{i}}^{y})}$$

(20)

$$d_{ij} = E(D_{y_{i}}^{\Delta_{m-1}}) + E(V_{ij}^{m})x$$

$$+ z_{\lambda}\sqrt{\text{Var}(D_{y_{i}}^{\Delta_{m-1}})} + \text{Var}(V_{ij}^{m})x^{2} + 2\eta x$$

(21)

$$\eta = \text{Cov}(V_{ij}^{m}, V_{ij}^{m})(\Delta_{n} - y_{i}) + \sum_{k=m+1}^{k=m-k} \text{Cov}(V_{ij}^{k}, V_{ij}^{k})(\Delta_{k}).$$

(22)

where $x = y - \Delta_{m-1}$, and $z_{\lambda}$ is the inverse CDF of the standard normal distribution at $\lambda$ confidence level. The quadratic equation, i.e., (21), can be easily solved as

$$x = \left(-a_{2} \pm \sqrt{a_{2}^{2} - 4a_{1}a_{3}}\right)/2a_{4}$$

(23)

$$a_{1} = z_{\lambda}^{2}\text{Var}(V_{ij}^{m}) - E(V_{ij}^{m})^{2}$$

(24)

$$a_{2} = 2z_{\lambda}\eta + 2d_{ij}E(V_{ij}^{m}) - 2E(D_{y_{i-1}}^{\Delta_{m-1}})E(V_{ij}^{m})$$

(25)

$$a_{3} = z_{\lambda}^{2}\text{Var}(D_{y_{i}}^{\Delta_{m-1}}) - (d_{ij} - E(D_{y_{i}}^{\Delta_{m-1}}))^{2}.$$  

(26)

Two solutions of $x$ can be found by (23). If $\lambda \geq 0.5$, then $x = (-a_{2} + \sqrt{a_{2}^{2} - 4a_{1}a_{3}})/2a_{4}$ is used; otherwise, $x = (-a_{2} + \sqrt{a_{2}^{2} - 4a_{1}a_{3}})/2a_{3}$ is chosen. Therefore, $\Phi_{T_{ij}}^{-1}(\lambda)$ can be calculated as

$$\Phi_{T_{ij}}^{-1}(\lambda) = x + \Delta_{m-1} - y_{i}.$$  

(27)

Based on the preceding method, the link travel time distribution can be generated using the following procedure.

**Procedure: GenerateLinkTime**

**Inputs:** Time instance $y_{i} \in (\Delta_{n-1}, \Delta_{n})$ and discrete number $L_{1}$

**Returns:** Discrete link travel time $\Phi_{T_{ij}}^{-1}(\lambda)$, $\lambda = \varepsilon_{1}, 2\varepsilon_{1}, \ldots, L_{1}\varepsilon_{1}$

**Step 1. Initialization**

Set time instance $\Delta_{m-1} := y_{i}$, mean distance $E(D_{y_{i-1}}^{\Delta_{m-1}}) := 0$, and variance of travel distance $\text{Var}(D_{y_{i-1}}^{\Delta_{m-1}}) := 0$.

Set $\Delta_{m} := \Delta_{n}$, and calculate $E(D_{y_{i}}^{\Delta_{m-1}}) := E(V_{ij}^{m})(\Delta_{n} - y_{i})$ and $\text{Var}(D_{y_{i}}^{\Delta_{m-1}}) := \text{Var}(V_{ij}^{m})(\Delta_{n} - y_{i})^{2}$.

Set $\beta_{m-1} := 0$ and $\eta := 0$, and calculate $\beta_{m} := 1 - \Phi_{D_{y_{i}}^{\Delta_{m}}}(d_{ij})$.

**Step 2. Generate discrete link travel times.**

For every $\lambda \in (\beta_{m-1}, \beta_{m})$

Calculate parameters $a_{1}, a_{2},$ and $a_{3}$ using (24)-(26).

If $\lambda \geq 0.5$, then $x = (-a_{2} - \sqrt{a_{2}^{2} - 4a_{1}a_{3}})/2a_{4}$; otherwise, $x = (-a_{2} + \sqrt{a_{2}^{2} - 4a_{1}a_{3}})/2a_{3}$.

Calculate $\Phi_{T_{ij}}^{-1}(\lambda) := \Delta_{m-1} - x - y_{i}$ using (27).

**End for**

**Step 3. Scan next time interval.**

Set $\Delta_{m-1} := \Delta_{m}$, $\Delta_{m} := \Delta_{m} + \Delta$, $E(D_{y_{i}}^{\Delta_{m-1}}) := E(D_{y_{i}}^{\Delta_{m}})$ and $\text{Var}(D_{y_{i}}^{\Delta_{m-1}}) := \text{Var}(D_{y_{i}}^{\Delta_{m}})$.

Calculate $\eta := \text{Cov}(V_{ij}^{m}, V_{ij}^{m})(\Delta_{n} - y_{i}) + \sum_{k=m+1}^{k=m-k} \text{Cov}(V_{ij}^{k}, V_{ij}^{k})(\Delta_{k})$.

Calculate $E(D_{y_{i}}^{\Delta_{m}}) = E(D_{y_{i}}^{\Delta_{m-1}}) + E(V_{ij}^{m})(\Delta_{n} - y_{i})$ and $\text{Var}(D_{y_{i}}^{\Delta_{m-1}}) := \text{Var}(D_{y_{i}}^{\Delta_{m-1}}) + \text{Var}(V_{ij}^{m})\Delta_{n}^{2} + 2\eta\Delta$.

Set $\beta_{m-1} := \beta_{m}$, and calculate $\beta_{m} := 1 - \Phi_{D_{y_{i}}^{\Delta_{m}}}(d_{ij})$.

If $\beta_{m-1} \geq 0.999$, then Stop; otherwise, goto Step 2.

The detailed steps in calculating the link travel time distribution (or arrival time distribution) using the S-TSM in Fig. 1 are given for illustration. The number of discrete elements is set to $L_{1} = 100(\varepsilon_{1} = 0.01)$. Consider a vehicle entering the link at $y_{1} = 8:00:10$. In Step 1, the travel distance distribution, i.e., $D_{y_{i}}^{\Delta_{i}}$, at $\Delta_{1} = 8:01:00$ can be calculated using (12)-(14). The mean and the variance of the travel distance are $E(D_{y_{i}}^{\Delta_{i}}) = 150$ and $\text{Var}(D_{y_{i}}^{\Delta_{i}}) = 900$. Then, $\beta_{0} = 0$ and $\beta_{1} = 1 - \Phi_{D_{y_{i}}^{\Delta_{i}}}(600) = 0$. Step 2 in the first iteration can be skipped because $\forall \lambda \notin (\beta_{0}, \beta_{1})$. In Step 3, the next time instance is set as $\Delta_{2} = 8:02:00$, and travel distance $D_{y_{i}}^{\Delta_{2}}$ can be calculated as $\eta = 15, E(D_{y_{i}}^{\Delta_{2}}) = 450$, and $\text{Var}(D_{y_{i}}^{\Delta_{2}}) = 6300$. Then, $\beta_{2} = 1 - \Phi_{D_{y_{i}}^{\Delta_{2}}}(600) = 0.029 < 0.999$, and therefore, the algorithm goes to Step 2 for the second iteration.

In Step 2 of the second iteration, there are two discrete points $\lambda = 0.01, 0.02 \in (0, 0.029)$. For $\lambda = 0.01$, three parameters, namely, $a_{1} = -19.6, a_{2} = 4661.7$, and $a_{3} = -197650.1$, can be calculated using (24)-(26). Because $\lambda = 0.01 < 0.5$, $x = (-a_{2} + \sqrt{a_{2}^{2} - 4a_{1}a_{3}})/2a_{3} = 55.2$ is calculated, and therefore, $\Phi_{T_{ij}}^{-1}(y_{1})(0.01) = 105.2$. Using the same method, the inverse CDF, i.e., $\Phi_{T_{ij}}^{-1}(y_{1})(0.02) = 108.2$, can be also calculated.
for $\lambda = 0.02$. In Step 3, the time instance is $\Delta_3 = 8:03:00$, and the travel distance distribution can be calculated as $\eta = 54$, $E(D^{T_{s,i}}_{y_{1}}) = 810$, and $\text{Var}(D^{T_{s,i}}_{y_{1}}) = 17964$. Then, $\beta_3 = 0.942 < 0.999$ can be obtained, and the algorithm goes to Step 2 for the third iteration.

As the algorithm progresses, $\Phi^{-1}_{T_{y_{1}}^{i}}(\lambda) \ \forall \lambda \in (0.029, 0.942)$ and $\forall \lambda \in (0.942, 0.999)$ can be respectively calculated in the third and fourth iterations. This way, the link travel time distribution $T_{i,j}(y_{1})$, as well as the arrival time distribution can be generated. The generated arrival time distribution (S-TSM($y_{1}$)) is illustrated in Fig. 1(c).

IV. SOLUTION ALGORITHM FOR FINDING THE LEAST EXPECTED TIME PATH

This section describes an efficient multicriteria A* algorithm for finding the least expected time path in a S-FIFO network, where all link travel time distributions satisfy the S-FIFO property. It can be proved as follows that the path travel time in the S-FIFO network satisfies the S-FIFO property.

Proposition 2: Given a path $p_{v}^{i}$ in the S-FIFO network network, the path travel time satisfies the S-FIFO property $y_{1} < y_{2} \Rightarrow \Phi^{-1}_{T_{y_{1}}^{i}}(\lambda) = y_{1} + \Phi^{-1}_{T_{y_{2}}^{i}}(\lambda) < \Phi^{-1}_{T_{y_{1}}^{i}}(\lambda) = y_{2} + \Phi^{-1}_{T_{y_{2}}^{i}}(\lambda)$, $\forall \lambda \in (0, 1)$.

Proof: See Chen et al. [22, Proposition 1]. $\Box$

Proposition 2 implies that, at any confidence level $\lambda$, travelers in the S-FIFO network cannot arrive at the destination earlier by departing later. This proposition enables the use of the following FSD [17] to eliminate dominated subpaths.

Definition 2: Given two paths $p_{u}^{i} \neq p_{v}^{i} \in P^{i}$ from origin $r$ to the same node $i$, $p_{u}^{i}$ FSD dominates $p_{v}^{i}$ if they satisfy $\Phi^{-1}_{T_{y_{2}}^{i}}(\lambda) < \Phi^{-1}_{T_{y_{1}}^{i}}(\lambda)$, $\forall \lambda \in (0, 1)$.

Definition 3: A path $p_{u}^{i}$ is said to be an FSD nondominated path if path $p_{u}^{i}$ cannot be FSD dominated by any path $p_{v}^{i} \in P^{i}$.

Let $\oplus$ be a path concatenation operator. $p_{u}^{rw} = p_{u}^{i} \oplus p_{w}^{i}$ means that path $p_{u}^{rw}$ goes through subpaths $p_{u}^{i}$ and $p_{w}^{i}$.

Proposition 3: In the S-FIFO network, if $p_{u}^{i}$ FSD dominates $p_{v}^{i}$, then $p_{u}^{rw} = p_{u}^{i} \oplus p_{w}^{i}$ always FSD dominates $p_{v}^{rw} = p_{v}^{i} \oplus p_{w}^{i}$ for any path $p_{w}^{i}$.

Proof: See Chen et al. [22, Proposition 2]. $\Box$

Based on Proposition 3, the least expected time path problem can be solved by using a generalized dynamic programming technique to determine a set of FSD nondominated paths. All FSD dominated paths, however, can be discarded without further consideration because they cannot be parts of the least expected time path.

It can be also proved as follows that the path travel time in the S-FIFO network is monotonically increasing with path extension.

Proposition 4: Given a path $p_{u}^{i}$ and a path $p_{v}^{j} = p_{u}^{i} \oplus a_{ij}$, their path travel time distributions $T_{u}^{j}(y_{r})$ and $T_{v}^{j}(y_{r})$ satisfy $\Phi^{-1}_{T_{y_{2}}^{j}}(\lambda) < \Phi^{-1}_{T_{y_{1}}^{j}}(\lambda)$, $\forall \lambda \in (0, 1)$.

Proof: See Chen et al. [22, Proposition 3]. $\Box$

With Proposition 4, the following two lemmas exist.

Lemma 1: Given a path $p_{u}^{i}$ and a path $p_{v}^{j} = p_{u}^{i} \oplus a_{ij}$, their expected path travel times satisfy $E(T_{u}^{j}(y_{r})) < E(T_{v}^{j}(y_{r}))$.

Proof: This can be easily deduced from Proposition 4. $\Box$

Lemma 2: The FSD nondominated path is acyclic in the S-FIFO network.

Proof: Suppose that $p_{u}^{i} \neq p_{v}^{i} \in P^{i}$ are two nondominated paths; that $p_{u}^{i}$ is acyclic, and that $p_{v}^{j} = p_{u}^{i} \oplus p_{w}^{i}$ passes through the same subpath $p_{u}^{i}$ but is containing one cycle subpath $p_{w}^{i}$ starting and ending at node $i$. According to Proposition 3, we have $\Phi^{-1}_{T_{y_{2}}^{j}}(\lambda) < \Phi^{-1}_{T_{y_{1}}^{j}}(\lambda)$, $\forall \lambda \in (0, 1)$. Therefore, $p_{v}^{j}$ FSD dominates $p_{u}^{i}$. This contradicts the assumption that $p_{u}^{i}$ and $p_{v}^{j}$ are two nondominated paths. $\Box$

Lemma 1 indicates that the objective function of the least expected time path finding problem is monotonically increasing with path extension, whereas Lemma 2 means that the FSD nondominated path is acyclic and does not pass through the same node more than once.

Based on these two lemmas, a multicriteria A* algorithm is proposed to efficiently determine the least expected time path without generating all FSD nondominated paths. The proposed algorithm makes use of a heuristic evaluation function as a label for path $p_{u}^{i}$, i.e.,

$$F(p_{u}^{i}) = E(T_{u}^{i}(y_{r})) + h(i) \quad (28)$$

where $h(i)$ is the estimated mean path travel time from node $i$ to destination $s$. The heuristic function is admissible if the following inequality is satisfied:

$$F(p_{u}^{i}) = E(T_{u}^{i}(y_{r})) + h(j) \geq F(p_{v}^{i}) = E(T_{v}^{i}(y_{r})) + h(i). \quad (29)$$

Equation (29) indicates that $F(p_{u}^{i})$ should monotonically increase with the path extension. A simple admissible $h(i)$ is the network distance function

$$h(i) = d_{is}/v_{\text{max}} \quad (30)$$

where $d_{is}$ is the shortest network distance from node $i$ to destination $s$, and $v_{\text{max}}$ is the maximum travel speed (or design speed) of a network.

Unlike the traditional time-dependent A* algorithm [9], in this proposed multicriteria A* algorithm, several FSD nondominated paths may need to be stored at each node when searching for the least expected time path. The detailed steps of the proposed multicriteria A* algorithm (named LET-A*) are described below.

Let $P^{i} = \{p_{1}^{i}, \ldots, p_{n}^{i}\}$ be a set of FSD nondominated paths from the origin $r$ to node $i$. All nondominated paths (from all nodes) are maintained in a scan eligible set, which is denoted by $SE = \{p_{1}^{i}, \ldots, p_{n}^{i}, \ldots, p_{n}^{j}, \ldots\}$ using a priority queue. All nondominated paths in the queue are ordered based on $F(p_{u}^{i})$. In each iteration, the label $p_{u}^{i}$ with minimum $F(p_{u}^{i})$ is selected from $SE$ for the path extension. An acyclic temporary path, which is denoted by $p_{u}^{i} := p_{u}^{i} \oplus a_{ij}$, is then constructed by extending the selected path $p_{u}^{i}$ to each successor node $j \in \text{SCS}(i)$. The temporary path $p_{u}^{i}$ is inserted into $P^{j}$ (the set of nondominated paths at node $j$) if $p_{u}^{i}$ is not FSD dominated by any path in $P^{j}$. The temporary path $p_{u}^{i}$ may also dominate paths in $P^{j}$. These dominated paths are discarded without further
considerations. The algorithm continues the path extension process until the destination is reached or \( SE \) becomes empty.

**Algorithm: LET-A**

**Inputs:** Origin and destination nodes and departure time \( y_r \)

**Returns:** The least expected time path

**Step 1. Initialization:**
- Create a path \( p_u^{i*} \) from origin to itself.
- Set arrival time distribution \( Y_r(\cdot) := 0 \) for the path \( p_u^{i*} \).
- Calculate \( h(r) \) and \( F(p_u^{i*}) \).
- Set \( P^r := \{ p_u^{i*} \} \) and \( SE := \{ p_u^{i*} \} \).

**Step 2. Path selection:**
- If \( SE = \varnothing \), then stop; otherwise, continue.
- Select \( p_u^{i*} \) at the top of \( SE \) and remove \( p_u^{i*} \) from \( SE \).
- If \( i = s \), then stop; otherwise, continue.

**Step 3. Path extension:**
- For every successor node \( j \in SC\{i\} \):
  - If \( j \notin \Phi_u^{i*} \), then continue; otherwise, scan next successor node.
  - Construct a temporary path \( p_u^{ij} := p_u^{i*} + a_{ij} \).
  - Call \( GeneratePathTime \) to generate arrival time distribution \( Y_j(\cdot) \) for the temporary path \( p_u^{ij} \).
  - Calculate \( h(j) \) and \( F(p_u^{ij}) \).
  - If \( p_u^{ij} \) is a FSD nondominated path, then insert \( p_u^{ij} \) into \( P^s \) and \( SE \) and remove all paths dominated by \( p_u^{ij} \) from \( P^s \) and \( SE \).

End for

Goto Step 2.

**Procedure: GeneratePathTime**

**Inputs:** Discrete arrival time distribution \( \Phi_{Y_i(\cdot)}(\lambda), \lambda = \varepsilon_2, 2\varepsilon_2, \ldots, L_2\varepsilon_2 \) for path \( p_u^{i*} \)

**Returns:** Discrete arrival time distribution \( \Phi_{Y_j(\cdot)}(\lambda), \lambda = \varepsilon_2, 2\varepsilon_2, \ldots, L_2\varepsilon_2 \) for path \( p_u^{ij} \)

**Step 1.** Generate arrival time distribution:
- For each element \( y_m^i \in \Phi_{Y_i(\cdot)}(\lambda) \) \( (y_m^i \in [\Delta_{m-1}, \Delta_m)] \)
  - Set \( y_m := (\Delta_{m-1} + \Delta_m)/2 \)
- If \( T_{ij}(y_m) = \varnothing \), then continue;
- Call \( GenerateLinkTime \) to generate discrete link travel time distribution \( \Phi_{T_{ij}(\cdot)}(\lambda), \lambda = \varepsilon_1, 2\varepsilon_1, \ldots, L_1\varepsilon_1 \)

End If

- For each discrete element \( t_{ij}^m \in \Phi_{T_{ij}(\cdot)}(\lambda) \)
  - \( y_j^k := y_i^m + t_{ij}^m \)

End for

End for

Sort all \( y_j^k \) in an ascending order by quick sort method

**Step 2.** Aggregate the number of discrete elements:
- For \( m = 1 \) to \( L_2 \)
  - \( \Phi_{Y_j(\cdot)}(m\varepsilon_2) := \frac{\sum_{k=mL_1}^{k=(m-1)L_1+1} y_j^k}{L_1} \)

End for

**Proposition 5:** If the heuristic function is admissible, the LET-A algorithm can determine the least expected time path when path \( p_u^{i*} \) is selected from \( SE \).

**Proof:** Suppose that \( P^s \) contains all nondominated paths between origin and destination nodes, except for \( p_u^{i*} \). Because path \( p_u^{i*} \) is selected from \( SE \), its heuristic function value \( F(p_u^{i*}) \) is less than that of any path in \( SE \). If the heuristic function is admissible, the \( F(p_u^{i*}) \) value is monotonically increasing with the path extension. Because all paths in \( P^s \) are extended from paths in \( SE \), \( F(p_u^{i*}) \) is less than the \( F(p_u^{i*}) \) value of any path in \( P^s \). Since \( h(s) = 0 \), \( F(p_u^{i*}) \) is equal to the mean path travel time, and therefore, \( p_u^{i*} \) is the least expected time path with a minimum expected path travel time. □

The complexity of the LET-A algorithm mainly depends on the used \( h(i) \) function and parameters \( L_1 \) and \( L_2 \). When \( h(i) = 0 \) is used, the LET-A algorithm reduces to a label-setting algorithm. In the worst case, the \( GenerateLinkTime \) procedure runs in \( O(|L_1|) \), and the \( GeneratePathTime \) procedure runs in \( O(L_1L_2\log(L_1L_2)) \). The label selection step (in the main procedure) requires \( O(|N||P|\log(|N||P|)) \) with an implementation of \( SE \) using a F-heap data structure [24], where \( |N| \) is the number of network nodes, and \( |P| \) is the maximum number of nondominated paths at a network node. The path extension step (in the main procedure) is \( O(|A||P|^2L_1L_2\log(L_1L_2)) \), where \( |A| \) is the number of network links. Therefore, the LET-A algorithm runs in \( O(|A||P|^2L_1L_2\log(L_1L_2) + |N||P|\log(|N||P|)) \). Theoretically, the LET-A algorithm has nonpolynomial complexity because \( |P| \) exponentially grows with the network size. In practice, several authors have found that the number of nondominated paths is much smaller than the maximum possible size [25], [26].

**V. NUMERICAL EXPERIMENTS**

**A. Case Study**

A simple network, which was presented by Sung et al. [7], is adopted in this paper to illustrate the S-TSM and the proposed LET-A algorithm. As shown in Fig. 2, the example network consists of five nodes and seven links. The travel speed distributions of each link in this example network are shown in Table I. Because all travel speed distributions are assumed to follow normal distributions, only their mean and coefficients of variation are given. The travel speed correlation coefficients for each two time intervals of each link are set to 0.3.

As aforementioned in Fig. 1, travel time distributions in the S-LTM follow a reciprocal of normal distribution and can be approximated as a lognormal distribution. Table II shows the means of the link travel time distributions, which are derived by dividing the link distance by corresponding mean travel speeds. The coefficients of variation of link travel time distributions in

![Example network (all link lengths: 10 km).](image-url)
the S-LTM are the same as those of the travel speed distributions in the S-TSM.

Fig. 3 illustrates the least expected time path finding results in the example network using both the S-TSM and the S-LTM. The origin and destination nodes were set as Nodes 1 and 5, respectively. It is shown in Fig. 3(a) that the path travel times under the S-LTM could violate S-FIFO consistency. For example, two vehicles departing at 10 and 20 min, respectively, choose the same path $p_{15}^{12} = a_{13} \oplus a_{35}$. As shown in Fig. 3(a), these two arrival time distributions (i.e., $Y_{15}^{12}(10)$ and $Y_{15}^{12}(20)$) intersect, and $\Phi_{Y_{15}^{12}(10)}^{-1}(\lambda) > \Phi_{Y_{15}^{12}(20)}^{-1}(\lambda), \forall \lambda \in (0.07, 1)$. In other words, using the same path $p_{15}^{12}$, the vehicle departing 10 min earlier will arrive at the destination later for the confidence interval $\lambda \in (0.07, 1)$. Fig. 3(b) shows the least expected time path finding results using the S-TSM. Clearly, these two arrival time distributions (i.e., $Y_{15}^{12}(10)$ and $Y_{15}^{12}(20)$) under the S-TSM satisfy the S-FIFO property because $\Phi_{Y_{15}^{12}(10)}^{-1}(\lambda) < \Phi_{Y_{15}^{12}(20)}^{-1}(\lambda), \forall \lambda \in (0, 1)$.

Comparing with Fig. 3(a) and (b), it is apparent that arrival times under the S-LTM may have a significant travel time bias. For instance, using the S-LTM, the expected arrival time is $E(Y_{15}^{12}(10)) = 51.3$ min, which is 14.5 min larger than that obtained by the S-TSM ($E(Y_{15}^{12}(10)) = 36.8$ min). It is also apparent in Fig. 3(a) and (b) that using the S-LTM, it is possible to misidentify a suboptimal path due to this S-LTM travel time bias. For example, for a vehicle departing from the origin at 30 min, the identified the least expected time path is $p_{15}^{15} = a_{13} \oplus a_{35}$ when the S-LTM is used. However, when the S-TSM is used, the actual least expected time path $p_{2}^{15} = a_{12} \oplus a_{24} \oplus a_{45}$ is found.

Fig. 3. Least expected time path finding results. (a) S-LTM. (b) S-TSM.
B. Computational Performance

This section reports the computational performance of the proposed \(LET-A^*\) algorithm in several large-scale road networks. The proposed algorithm was coded in the Visual C# programming language. The priority queue was implemented using the F-heap data structure [24]. Link travel time distributions were discretized into \(L_1 = 20\) intervals, and arrival time distributions at each node were discretized into \(L_2 = 100\) intervals.

To perform a comparative evaluation of the \(LET-A^*\) algorithm, the multicriteria label-correcting algorithm (called the \(LET-EV\) algorithm) proposed by Miller-Hooks and Mahmassani [14] was also implemented using the same programming language. In the \(LET-EV\) algorithm, each link travel time was discretized into 20 intervals. All experiments were conducted on a computer with a four-core Intel i3-2100 central processing unit running at 3.1 GHz (only one core was used) and the Windows 7 operating system.

For computational tests, the travel speed estimates from a real-world advanced traveler information system in Hong Kong were collected. In Hong Kong, travel speed estimates on major urban roads are provided by Real-Time Travel Information System (RTIS) at 5-min intervals [27] (http://tis.td.gov.hk/rtis/rtis/index/main_partial.jsp). In this paper, RTIS travel speed estimates on Wednesday for year 2009 were collected to generate travel speed distributions for each link in each 5-min interval. Using the collected data, the means and variances of the travel speed distributions were calculated, and the travel speed correlations for each link between every pair of time intervals were also generated.

In addition to the RTIS network, the well-known “Sioux Falls” and “Chicago Regional” road networks were used in the computational tests. For these two networks, the travel speed distributions of each link were constructed by randomly selecting travel speed distributions of one RTIS link. The characteristics of these three test networks are summarized in Table III.

The \(LET-A^*\) and \(LET-EV\) algorithms were then used to determine the least expected time path in these three networks. In the \(LET-EV\) algorithm, the departure times were discretized using 1-min intervals, and only the morning period (6:30 A.M. to 10:30 A.M.) was considered. In the computational tests, departure times were set to 8 A.M. All reported results were the average of 100 runs, using different origin–destination pairs for each run. The 100 origin–destination pairs were randomly selected for each network, and the same set of pairs was used for every test performed on a given network. Computational performance was evaluated in terms of computing time and the number of generated nondominated paths in the testing network.

The computational performance of the \(LET-A^*\) and \(LET-EV\) algorithms is given in Table IV. It is apparent that the proposed \(LET-A^*\) algorithm runs significantly faster than the \(LET-EV\) algorithm. For example, in the RTIS network, the \(LET-A^*\) algorithm was about 500 (54.73/0.11) times faster than the \(LET-EV\) algorithm. This result was to be expected because the \(LET-A^*\) algorithm generated far fewer nondominated paths than the \(LET-EV\) algorithm. For example, in the RTIS network, the number of nondominated paths generated by the \(LET-A^*\) algorithm was 247 times (88 014/346) fewer than that generated by the \(LET-EV\) algorithm.

Compared with the \(LET-EV\) algorithm, this reduction of nondominated paths in the \(LET-A^*\) algorithm is mainly due to the following two factors. First, the \(LET-EV\) algorithm uses the S-LTM to represent STD link travel times, which may violate the S-FIFO property. In this case, all EV nondominated paths for all possible departure times must be generated. In contrast to the \(LET-EV\) algorithm, link travel times in the \(LET-A^*\) algorithm are based on the S-TSM and, therefore, are guaranteed to be S-FIFO consistent. With this S-FIFO property of link travel times, only FSD nondominated paths need to be generated for the given departure time (i.e., 8 A.M.), and therefore, the number of generated nondominated paths can be significantly reduced in the \(LET-A^*\) algorithm.

Second, the \(LET-EV\) algorithm uses the label-correcting search type, and thus, the path search process cannot be stopped until all EV nondominated paths from the origin to all other network nodes have been generated. The \(LET-A^*\) algorithm, however, uses the A* search process, and the search process can be terminated as soon as the first FSD nondominated path has been selected from the scan eligible. Therefore, the \(LET-A^*\) algorithm can determine the least expected time path without generating all FSD nondominated paths from the origin to all other nodes. This A* search process also contributes to the reduction of generated nondominated paths in the \(LET-A^*\) algorithm.

Table IV also shows that the computing time of the \(LET-A^*\) algorithm grows almost linearly with network size. For example, the computing time required by the \(LET-A^*\) algorithm in the Sioux Falls network with 24 nodes was 0.003 s. When the RTIS network was analyzed, the number of nodes was increased approximately 57 times (1367/24) and the computing time approximately 37 times (0.11/0.003). In contrast to the \(LET-A^*\) algorithm, the computational performance of the \(LET-EV\) algorithm exponentially degraded with the network size. For example, when the RTIS network was analyzed, the computing time required by the \(LET-EV\) algorithm significantly increased by about 2488 times (54.73/0.022). This degradation of the \(LET-EV\) algorithm occurred because the number of
generated EV nondominated paths exponentially increased with the network size. It is shown in Table IV that when the RTIS network was used, the number of generated nondominated path increased by 1851 times (88,014/47.54).

VI. Conclusion

This paper has addressed the problem of finding the least expected time path in STD networks, where link travel time distributions vary with the time of day. The S-FIFO property of link travel times in STD networks was investigated. The S-FIFO property is a stochastic extension of the traditional FIFO principle. S-FIFO implies that at any confidence level, a vehicle departs from the origin earlier cannot arrive at the destination later. This S-FIFO property enables the use of a generalized dynamic programming technique in solving the least expected time path finding problem, by determining a set of nondominated paths under the FSD.

It was found that link travel times in the STD network satisfy the S-FIFO property. The violation of the S-FIFO property in the traditional S-LTM is due to the assumption that the link travel speed of a vehicle is fixed while the vehicle is traveling on the link. This assumption can lead to significant travel time bias when traffic conditions dramatically change. In this paper, this assumption was relaxed by proposing a new S-TSM. In the S-TSM, the travel speed distribution of a vehicle depends on the time instance when the vehicle enters that link, and travel speed distributions of the vehicle can be varied while the vehicle is traveling on the link. A discrete approximation method was proposed in this paper to generate the link travel time distribution from travel speed distributions. It was rigorously proved that the link travel time distributions in the S-TSM satisfy the S-FIFO property.

Based on the proposed S-TSM, a multicriteria A* algorithm was proposed to exactly determine the least expected time path. Computational tests using several large-scale road networks were conducted. Numerical results showed that the proposed algorithm has a significant computational advantage over the existing multicriteria label-correcting algorithm built on the traditional S-LTM [14]. The results of the case study indicated that the proposed algorithm can determine the least expected time path in large-scale road networks within a reasonable computing time and therefore have potential uses in real-world route guidance systems.

On the basis of the solution algorithm proposed in this paper, certain extensions can be envisaged.

1) In this paper, the $\alpha$-approximation method [23] was used to generate path travel time distributions. This $\alpha$-approximation method can obtain accurate path travel time distribution when the number of discrete elements is sufficiently large. This approximation method, however, requires a considerable computational effort. The development of efficient methods for generating path travel time distributions requires further investigation.

2) In the case study, only historical travel speed data were utilized to generate travel speed distributions. To provide a real route guidance service, accurate travel speed distributions can be predicted by fusing both historical and real-time travel speed data [13].

3) The proposed algorithm can determine the least expected time path for travelers in the face of travel time uncertainty. However, empirical studies have found that travelers under travel time uncertainty may consider not only the expected path travel time but also the probability of on-time arrival (or called travel time reliability) in their route choice decisions [28]–[30]. Further studies, therefore, are needed to extend the proposed solution algorithm to reliable shortest path problems [31]–[37] by considering travelers’ on-time arrival concerns. The established S-FIFO property of link travel time can be also very useful in the development of efficient solution algorithms for solving reliable shortest path problems.

REFERENCES


Bi Yu Chen (M’13) received the Bachelor’s degree in surveying and mapping engineering and the Master’s degree in geographical information science from Wuhan University, Wuhan, China, in 2003 and 2006, respectively, and the Ph.D. degree in transportation engineering from the Hong Kong Polytechnic University, Kowloon, Hong Kong, in 2012. He is currently a Lecturer and a Postdoctor Fellow with the State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University. He has authored or coauthored more than 30 papers in international journals and conferences. His research interests are in the areas of intelligent transportation systems, transport geography, and network reliability and vulnerability modeling.

William H. K. Lam received the Bachelor’s degree and the Master’s degree in civil engineering from the University of Calgary, Calgary, AB, Canada, in 1978 and 1980, respectively, and the Ph.D. degree in transportation engineering from the University of Newcastle upon Tyne, Tyne, U.K., in 1992. He is currently a Chair Professor of civil and transportation engineering and an Associate Head of the Department of Civil and Environmental Engineering with The Hong Kong Polytechnic University, Kowloon, Hong Kong. He has authored or coauthored more than 200 journal papers together with 50 consultancy reports. His research interests include transport network modeling and infrastructure planning, travel demand forecasting and risk assessment, intelligent transportation system technology and planning, and public transport and pedestrian studies.

Qingquan Li received the M.S. degree in engineering and the Ph.D. degree in photogrammetry and remote sensing from Wuhan University, Wuhan, China, in 1988 and 1998, respectively. He is currently a Professor with the State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University; the President of Shenzhen University, Shenzhen, China; and the Director of the Engineering Research Center for Spatio-Temporal Data Smart Acquisition and Application, Ministry of Education of China.

Agachai Sumalee received the Bachelor’s degree in civil engineering from King Mongkut’s Institute of Technology Ladkrabang, Bangkok, Thailand, and the Master’s degree and the Ph.D. degree in transport planning and engineering from the University of Leeds, Leeds, U.K.

Ke Yan is an undergraduate student with the International School of Software, Wuhan University, Wuhan, China.

He is currently a Research Assistant under the supervision of Dr. Bi Yu Chen. His research interests include cloud computing, large-scale data processing, and information retrieval.

Prof. Lam is the President of the Hong Kong Society for Transportation Studies. He is also the Editor-in-Chief of Transportmetrica and a Coeditor-in-Chief of the Journal of Advanced Transportation.

Prof. Li is an Editorial Board Member of the Surveying and Mapping Journal and the Information Science Edition of Wuhan University Journal.