Constraint Reformulation and a Lagrangian Relaxation–based Solution Algorithm for a Least Expected Time Path Problem

Lixing Yang
State Key Laboratory of Rail Traffic Control and Safety, Beijing Jiaotong University, Beijing, 100044, China, lxyang@bjtu.edu.cn
Xuesong Zhou
Department of Civil and Environmental Engineering, University of Utah, Salt Lake City, UT 84112-0561, USA, zhou@eng.utah.edu

Abstract

Using a sample-based representation scheme to capture spatial and temporal travel time correlations, this article constructs an integer programming model for finding the a priori least expected time paths. We explicitly consider the unique path constraint associated with the a priori path in a time-dependent and stochastic network, and propose a number of reformulations to establish linear inequalities that can be easily dualized by a Lagrangian relaxation solution approach. The relaxed model is further decomposed into two sub-problems, which can be solved directly by using a modified label-correcting algorithm and a simple single-value linear programming method. The numerical experiments investigate the quality and computational efficiency of the proposed solution approach.

Key words: a priori least expected time path; time-dependent traffic network; Lagrangian relaxation

History:

1. Introduction

Finding the shortest path in a stochastic network is among the core network optimization problems in the fields of both operations research and transportation engineering. To mitigate recurring and non-recurring congestion, efficient routing algorithms are critically needed to minimize the expected path travel time in network-wide traffic flow management and personalized route guidance applications. Three categories of stochastic programming formulations have been proposed to represent the inherent travel time uncertainties in traffic networks: (1) discrete probability distribution functions (e.g., Miller-Hooks 1997, 2001; Nie and Wu 2009a), (2) moment-based characterization for continuous link travel times through variance and other statistics (e.g., Fu and Rilett 1998; Sun, Gu and Mahmassani 2011) and (3) a sample-based representation, which can be viewed as a special version of the Monte Carlo approximation method. The third representation method has been used by Liu (2002) to approximate service time window constraints for a vehicle routing problem, by Chen and Ji (2005) to simulate uncertain objectives for a stochastic path finding problem and, recently, by Xing and Zhou (2011) to capture spatial travel time correlations for the most reliable path problem.

In a transportation network with stochastic, time-varying link travel times, the optimal routing strategies can be evaluated according to various objective functions, including (1) finding the least expected travel time (Hall 1986; Miller-Hooks 2001; Fu 2001; Nielsen 2003; Yang and Miller-Hooks 2004; Gao and Chabini 2006; Hickman and Bernstein 1997; Sivakumar and Batta 1994), (2) maximizing the probability of arriving on time (Nie and Wu 2009a; Samaranayake et al. 2011) and (3) minimizing the maximum possible travel time (Nielsen 2003). In addition, the existing routing algorithms are usually used to provide either an a priori optimal path (as a single solution) or adaptive en-route path trees (as a set of solutions). In the a priori path problem, based on (predicted) stochastic travel times, travelers are required to select a single route before departing from the origin node; this selected path cannot be updated, even though only one instance of stochastic travel time will be realized. In adaptive en-route guidance applications, which can be viewed as a multi-stage recourse problem (Miller-Hooks 2001), travelers need to select their route at each node after new or updated travel time information is available. In other words, the selection of the next physical link will be executed after the link traversal times are revealed en route once the link is traversed.

To handle travel time stochasticity and dynamics in route selection, according to Miller-Hooks (2001), many studies use two major classes of solution strategies to choose the desirable paths, namely, adaptive selection and a priori optimization. The early work by Hall (1986) presents the terminology and solution technique for adaptively updating path solutions when seeking the least expected travel time path. Recently, Miller-Hooks (2001), Yang and Miller-Hooks (2004) and Gao and Chabini (2006) established more comprehensive frameworks for finding optimal routing policies or a set of strategies (i.e., hyperpaths) in
stochastic time-dependent networks. A study by Nielsen (2003) considers the problem of finding the K best strategies and a bicriterion route with time-adaptive strategies for route choice. In addition, focusing on a route travel time reliability measure, Samaranayake et al. (2011) designed an adaptive routing policy to maximize the probability of arriving on time at a destination.

Aiming to extend the single-valued dominance rule in the standard shortest path algorithm with constant link distance, many researchers focus on how to construct effective dominance rules to solve a priori optimization for the shortest path problem under dynamic and uncertain traffic conditions. For instance, an early study by Wellman, Ford and Larson (1995) enhanced the path planning algorithm based on the stochastic consistency and stochastic dominance principle. Miller-Hooks and Mahmassani (2000) defined an optimal a priori solution as the path that can realize the least expected travel time in the network. Moreover, in their later studies, these researchers further investigated a systematic approach to compare sub-path travel times (at intermediate nodes) and the optimality of solutions by constructing various dominance rules, including a deterministic dominance rule, a first-order stochastic dominance rule and an expected-value dominance rule (Miller, Mahmassani and Ziliaskopoulos 1994; Miller-Hooks 1997; Miller-Hooks and Mahmassani 2003). Based on a probability distribution function (PDF)-based representation, Miller-Hooks and Mahmassani (2000) presented a novel lower bound estimator by considering the least revealed travel times on traveled links en route through time-adaptive routing selection rules.

Focusing on an expected-value dominance rule, with the travel time on each arc being treated as a continuous time stochastic process, Fu and Rilett (1998) also developed probability-based methods to approximate the mean and variance of the travel time for a given path in a dynamic and stochastic network. Using the first-order stochastic dominance rule, Nie and Wu (2009a, b) and Wu and Nie (2009) proposed various methods of finding reliable a priori shortest paths to guarantee a given likelihood of arriving on time at the destination. Pretolani (2000) presented several criteria, such as the least expected travel time and minimization of the maximum possible travel time, to rank hyperpaths in a time-expanded directed hypergraph. Additionally, a variety of dominance rules have been embedded in a label-correcting algorithmic framework by Miller, Mahmassani and Ziliaskopoulos (1994), Miller-Hooks (1997), Nie and Wu (2009a), and Wu and Nie (2009). A route generation-oriented solution scheme, which is typically built on the k-shortest path algorithm, has been adapted by Fu and Rilett (1998), Wu et al. (2005) and Nielsen (2003).

In this research, we focus on the least expected travel time (LET) criterion, which can be easily estimated and validated within a traditional utility-maximization framework. We are particularly interested in how to find the optimal a priori path using a sample-based representation for time-dependent and stochastic travel times. This paper aims to offer the following contributions to the growing body of work on optimum path algorithms in dynamic network analysis.

1) A rigorous mathematical programming model is formulated for finding the optimal route with the least expected travel time problem, in which scenario-based time-dependent link travel times are used to capture possible spatial and temporal correlations. The resulting unique path constraint (for the a priori path) across different scenarios is then reformulated by different equivalent forms to allow efficient problem decomposition and constraint relaxation. Note that formulating or reformulating the stochastic routing problem has not received sufficient attention in the literature; this is because the widely studied adaptive routing problem is essentially a complex multi-stage decision process, and the PDF-based representation is difficult in its own right to express in a standard integer programming model.

2) Another emphasis of this paper is how to quantify the quality of the solution and provide benchmarks for evaluating various heuristic algorithms. A novel Lagrangian relaxation-based lower bound approach is developed to handle multiple stochastic scenarios and to iteratively approximate the optimal solution space, in which the dualized model can be further decomposed into two sub-problems that are easily solved by efficient label-correcting algorithms and simple rules for a univariate linear program.

In comparison with several related important studies on this topic, the main features of our paper are summarized in Table 1.
Table 1 Comparative Summary of Shortest Path Problems with Stochastic Travel Times

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Representation of stochastic input data</td>
<td>Time-varying PDF</td>
<td>Time-varying PDF</td>
<td>Samples with spatial correlation, static stochastic travel times</td>
<td>Scenario-based time-dependent link travel time with both spatial and temporal correlations</td>
</tr>
<tr>
<td>Objective function</td>
<td>Pareto optimal solutions</td>
<td>Local-reliable paths</td>
<td>Mean + standard deviation of path travel time</td>
<td>Expected path travel time</td>
</tr>
<tr>
<td>Solution methodology</td>
<td>Three dominance rules</td>
<td>First-order stochastic dominance rules</td>
<td>Variable substitution for non-additive objective function, Lagrangian relaxation</td>
<td>Unique path constraint reformulation, Lagrangian relaxation with multiple scenarios</td>
</tr>
<tr>
<td>Characteristic of solution</td>
<td>Upper bound, lower bound through adaptive route choice</td>
<td>Upper bound of objective function</td>
<td>Upper bound, lower bound through Lagrangian relaxation</td>
<td>Upper bound, lower bound through Lagrangian relaxation</td>
</tr>
</tbody>
</table>

The remainder of this paper is structured as follows. Section 2 provides a formal problem statement, followed by three versions of reformulated models for the original problem in Section 3. In Section 4, a Lagrangian relaxation method is adapted for seeking a tight lower bound of the objective function, and the corresponding relaxed model is then decomposed into two classes of sub-problems associated with each scenario and each physical link, respectively. Section 5 presents a design of a sub-gradient algorithm to iteratively improve the solution quality and reduce the optimality gap. Finally, several numerical experiments are conducted in Section 6 to demonstrate the computational effectiveness of the proposed algorithms.

Table 2 Subscripts and Parameters Used in Mathematical Formulations

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S$</td>
<td>= set of nodes in the physical traffic network.</td>
</tr>
<tr>
<td>$V$</td>
<td>= set of traffic links.</td>
</tr>
<tr>
<td>$T$</td>
<td>= set of discrete time stamps.</td>
</tr>
<tr>
<td>$t$</td>
<td>= index of different time stamps, $t \in {t_0, t_0 + \sigma, t_0 + 2\sigma, \ldots, t_0 + M\sigma}$.</td>
</tr>
<tr>
<td>$d$</td>
<td>= index of scenarios/days, $d \in {1, 2, \ldots, D}$.</td>
</tr>
<tr>
<td>$i, j$</td>
<td>= indices of nodes, $i, j \in S$.</td>
</tr>
<tr>
<td>$(i, j)$</td>
<td>= index of traffic link between adjacent nodes $i$ and $j$, $(i, j) \in V$.</td>
</tr>
<tr>
<td>$c_{ij}^d$</td>
<td>= travel time on traffic link $(i, j)$ at the entering time $t$ on scenario/day $d$.</td>
</tr>
<tr>
<td>$C^d$</td>
<td>= vector representation for sequence $c_{ij}^d, (i, j) \in V, t \in T$ on scenario/day $d$.</td>
</tr>
</tbody>
</table>
2. Problem Statements

Consider a directed, connected traffic network \((S, V)\), where \(S\) is a finite set of nodes, and \(V\) is a finite set of traffic links between different adjacent nodes. The planning time horizon is discretized into a set of small time slots, denoted by \(T = \{t_0, t_0 + \sigma, t_0 + 2\sigma, \cdots, t_0 + M\sigma\}\). Symbol \(t_0\) specifies the given departure time from the origin node \(O\), and \(\sigma\) represents a short time interval (e.g. 6 seconds) during which no perceptible changes of travel times are assumed to take place in a transportation network. \(M\) is a sufficiently large positive integer so that the time period from \(t_0\) to \(t_0 + M\sigma\) covers the entire planning horizon. Corresponding to each physical link \((i, j)\), a time-variant link travel time \(c^d_{ij}\) is given for traveling along the link when a vehicle departs from node \(i\) at each time stamp \(t\in T\) under scenario \(d\). Table 2 lists the notations for the least expected travel time shortest path problem under consideration.

2.1. Decision Variables

Two types of decision variables in Table 3 will be used to show the selection of physical links and time-dependent arcs.

<table>
<thead>
<tr>
<th>Table 3 Decision Variables Used in Mathematical Formulations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Symbol</td>
</tr>
<tr>
<td>(x_{ij})</td>
</tr>
<tr>
<td>(y^d_{ij})</td>
</tr>
<tr>
<td>(X)</td>
</tr>
<tr>
<td>(Y^d)</td>
</tr>
<tr>
<td>(Y)</td>
</tr>
</tbody>
</table>

2.2. Constraints on Networks

In the a priori LET path problem, the main task is to seek a unique physical route in the traffic network such that the expected total travel time over different random scenarios can be minimized. The physical network flow balance constraint is first introduced for the LET path selection process in the physical network \((S, V)\).

\[
\sum_{(i, j) \in V} x_{ij} - \sum_{(j, i) \in V} x_{ji} = \begin{cases} 
1, & i = O \\
-1, & i = D \\
0, & \text{otherwise}
\end{cases}.
\]  

(1)

The set of constraints ensures that all of the selected physical links can constitute a path from the origin \(O\) to the destination \(D\) on each scenario. Constraint (1) will be abbreviated using the linear form “\(AX = b\)”, as below.
This research aims to use a sampling-based approach to capture complex temporal and spatial travel time correlations in a traffic network; in particular, historical travel times over different days are used to construct scenarios. Note that the travel time correlations can be contributed by physical bottlenecks and a number of delay resources such as incidents, road construction, severe weather conditions and special events. On the other hand, in a pre-trip routing application involving an advanced travel time forecasting engine, each scenario can also correspond to a stochastic instance of predicted travel times under a certain incident duration or capacity reduction level.

Consider an illustrative network consisting of three nodes and three links in Figure 1. We need to select an a priori LET path from node 1 to node 3 under two scenarios of time-varying link travel times \( d = 1, 2 \), in which the departure time at the origin node is set as 1 minute. There are two potential paths, namely, P1: 1 → 2 → 3, and P2: 1 → 3 for the given OD pair. Figure 2 displays the corresponding schedules along those two paths, where the y-axis represents nodes on each path and the x-axis represents the planning time horizon. Then, the evaluation value of P1 (or P2) is the averaged travel time in cases of \( d = 1, 2 \) on path 1 → 2 → 3 (or 1 → 3). Table 4 displays the expected travel times on each route, in which no waiting is permissible at intermediate nodes. The expected travel times on P1 and P2 over two scenarios are 7 min and 7.5 min, respectively; thus, the least expected time path is found to be P1: 1 → 2 → 3.

### Table 4 Path Travel Time Data

<table>
<thead>
<tr>
<th>Route</th>
<th>Travel Time (d=1)</th>
<th>Travel Time (d=2)</th>
<th>Expected Travel Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 → 2 → 3 (optimal)</td>
<td>9</td>
<td>5</td>
<td>7</td>
</tr>
<tr>
<td>1 → 3</td>
<td>7</td>
<td>8</td>
<td>7.5</td>
</tr>
</tbody>
</table>

For each scenario \( d \), one can construct the corresponding space-time network, denoted by \( (S_d, V_d) \), which is expanded from the physical network \( (S, V) \) and time-varying link travel time. In more detail, \( S_d = \{i_t | i \in S, t \in T \} \) represents the set of time-dependent nodes, where \( i_t \in S_d \) indicates the state of node \( i \) at time stamp \( t \), and each state will be treated as a separated node. The set of time-dependent arcs is represented by \( V_d = \{(i_t, j_{t'}) | (i, j) \in V, t + c_{ij}^d = t', t_0 \leq t \leq t', t_0 + M \sigma \} \), in which time-dependent arc \((i_t, j_{t'})\) occurs in the space-time network when one can travel from physical node \( i \) at time stamp \( t \) and arrive at physical node \( j \) at stamp \( t' \), where \( t + c_{ij}^d = t', t_0 \leq t \leq t_0 + M \sigma \).
For each scenario-based space-time network, \( y_{ijt}^d \) is the binary variable indicating whether a time-dependent arc from node \( i \) to node \( j \) at starting time \( t \) is selected in the least travel time path under scenario \( d \). For the same departure time \( t_0 \) in a space-time network, there are different time-dependent paths with different arrival times at the destination. In order to construct standard flow balance constraints, a dummy destination node \( D_{t_0+M\sigma} \) can be added to the space-time network to represent the completion of the journey, in which the corresponding link travel times are set to zero for incoming links from the original destination \( D \) to the dummy destination node \( D_{t_0+M\sigma} \), where \( t' \) belongs to the feasible arrival times at the destination. Figure 3 gives an illustration for the added dummy destination with \( M = 3 \).

\[ \begin{align*}
O & \quad t_0 \\
D & \quad t_0 + \sigma \\
D & \quad t_0 + 2\sigma \\
D & \quad t_0 + 3\sigma \\
D_{t_0+3\sigma} & \quad D
\end{align*} \]

**Figure 3: An Illustration of Space-Time Network and Dummy Destination**

To ensure the feasibility of a time-dependent path, a space-time flow balance constraint should be formulated as follows:

\[
\sum_{(i,j) \in V_d} y_{ijt}^d - \sum_{(j,i) \in V_u} y_{jim}^d = \begin{cases} 
1, & i = O, \ t = t_0 \\
-1, & i = D', \ t = t_0 + M\sigma . \\
0, & \text{otherwise}
\end{cases}
\]

Equation (2) will be denoted below by an abbreviated form, “\( B^d Y^d = h^d \).”

Essentially, the goal of the a priori LET path problem is to find a path in the physical network that can correspond to multiple paths in individual scenario-dependent space-time networks with possible different arrival times at the destination and intermediate nodes. To characterize this critical linkage between the physical and space-time network representation, we introduce (space-time) arc-to-link mapping constraints to capture the relationship between the selection of a physical link and potential time-dependent arcs as follows:

\[
y_{ijt}^d \leq x_{ij}, \ (i,j) \in V, \ d = 1, 2, \ldots, D, t \in T .
\]

This “if-then” conditional constraint (widely used in integer programming formulations) ensures that a time-dependent arc \((i,j)\) can be selected if and only if a physical link \((i,j)\) lies on the final path. Constraint (3) will be referred to as “\( Y^d \leq X \).”

Finally, a binary variable constraint should be imposed on decision variables \( x_{ij} \) and \( y_{ijt}^d \). That is,

\[
x_{ij}, y_{ijt}^d \in \overline{B}, \forall (i,j) \in V, \ d = 1, 2, \ldots, D, t \in T .
\]

Constraint (4) will be abbreviated as “\( X, Y \in \overline{B} \).”

**2.3 Objective Function**

For any solution method, a natural choice for a robust a priori path is to select a solution \( X \) so as to minimize the expected total travel time in the scenario-based time-dependent travel data. Then the objective is
\[ F(X,Y) = \frac{C_Y}{D} = \frac{1}{D} \sum_{d=1}^{D} \sum_{(i,j) \in V} \sum_{t \in T} c_{ijt}^d y_{ijt}^d . \]

Because the number of scenarios \( D \) is a constant, without loss of generality, this equation is simplified as an equivalent objective in the following discussion. That is,

\[ F(X,Y) = CY . \] (5)

2.4. One-Stage Mathematical Model

By further introducing an objective that minimizes the expected value of least time-dependent travel times over all scenarios in equation (5), the integer programming model of the a priori LET problem can be constructed below:

\[
\begin{align*}
\min & \quad \text{expected total travel time (5)} \quad F(X,Y) = CY \\
\text{s.t.} & \quad \text{physical network flow balance constraint (1)} \quad AX = b \\
& \quad \text{space-time flow balance constraint for each scenario(2)} \quad B^d Y^d = h^d, \quad d = 1, 2, \ldots, D \\
& \quad \text{space-time arc-to-link mapping constraint (3)} \quad Y^d \leq X, \quad d = 1, 2, \ldots, D . \\
& \quad \text{binary variable constraint (4)} \quad X, Y \in \bar{B}.
\end{align*}
\] (6)

2.5. Alternative Two-Stage Mathematical Model

Adaptive routing is recognized as an alternative method to search for the least expected time path in a stochastic and time-dependent network, which can be treated using a two-stage optimization problem with recourse. Specifically, the process of a routing selection can be divided into two steps. The first step is to choose a sub-path from the origin to some intermediate node in a sub-area of the network. Then, the rest of the sub-path, which connects the aforementioned intermediate node and the destination, is determined by the least expected time criterion. Meanwhile, the corresponding least expected time will be used to evaluate this path selection.

Note that in this paper we focus the first stage on entire routing selection; sub-area-based route selection at the first stage will be discussed in future papers. The stochastic two-stage optimization model with recourse can be formulated as follows:

\[
\begin{align*}
\min & \quad E[Q(X,d)] \\
\text{s.t.} & \quad AX = b \\
& \quad X \in \bar{B} \\
& \quad \text{where } Q(X,d) = \min C^d Y^d \\
& \quad \text{s.t. } B^d Y^d = h^d \\
& \quad Y^d \leq X \\
& \quad Y \in \bar{B} .
\end{align*}
\] (7)

where \( C^d Y^d = \sum_{(i,j) \in V} \sum_{t \in T} c_{ijt}^d y_{ijt}^d \). In the above model, the decision variable \( X \) at the first stage is used for selecting physical routes in the transportation network. At the second stage, the least total travel time associated with the space-time network on each scenario needs to be calculated. The recourse objective function of the first stage, which is essentially the same as the objective in model (6), represents the least expected time-dependent travel time over different scenarios. Note that, although model (7) can well address the problem mathematically, some inherent difficulties remain in solving multi-stage models. For this reason, the emphasis of this paper will be on model (6).

3. Reformulating Complex Mapping Constraints

Within a Lagrangian relaxation framework, the flow balance constraints (1) and (2) can be viewed as easy constraints because they can be handled explicitly in standard shortest path algorithms. The space-time arc-to-link mapping constraint (3) then becomes a set of hard constraints. The following discussion focuses
on how to convert this hard constraint set for each scenario.

3.1. Introducing Scenario-Dependent Physical Link Indicators

Recall that the path solution to the original LET model should be unique across different scenarios, and this requirement is enforced by constraints (1) and (3). We further introduce a unique path constraint as the first version of model reformulation. To this end, an alternative set of decision variables is needed, as shown in Table 5.

Table 5 New Decision Variables Used in Reformulations

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x^d_{ij}$</td>
<td>= 1 if traffic link $(i, j)$ is selected on scenario/day $d$; 0 otherwise.</td>
</tr>
<tr>
<td>$X^d$</td>
<td>= vector representation for sequence $x^d_{ij}, (i, j) \in V$ on scenario/day $d$.</td>
</tr>
<tr>
<td>$X$</td>
<td>= vector representation for sequence $x^d_{ij}, (i, j) \in V, d = 1, 2, \ldots, D$.</td>
</tr>
</tbody>
</table>

It is worth noting that, without a specific statement, the notation “$X$” mentioned below always has the meaning shown in Table 5. With new decision variable $x^d_{ij}$, the physical network flow balance constraint (1) and space-time arc-to-link mapping constraint (3) will be reformulated as follows:

$$
\sum_{(i, j) \in V} x^d_{ij} - \sum_{(j, i) \in V} x^d_{ji} = \begin{cases} 
1, & i = O \\
-1, & i = D, \ d = 1, 2, \ldots, D, \\
0, & \text{otherwise}
\end{cases},
$$

multi-equations-based unique path constraint: $x^d_{ij} = x^{d-1}_{ij} = \cdots = x^{d=1}_{ij}, (i, j) \in V$. \hspace{1cm} (9)

$$
y^d_{ij} \leq x^d_{ij}, (i, j) \in V, d = 1, 2, \ldots, D, t \in T. \hspace{1cm} (10)
$$

For notational convenience, constraints (8), (9) and (10) are abbreviated as vector-based formulations by “$AX^d = b$, $d = 1, 2, \ldots, D$”, “$X^d = X^{d-1} = \cdots = X^{d=1}$”, and “$Y^d \leq X^d$, $d = 1, 2, \ldots, D$”, respectively. Substituting constraints (1) and (3) in model (6) by constraints (8) through (10), we can generate an alternative version of the original problem.

3.2. Transforming Multiple Equalities to Nonlinear Variance Equality

Although the model mentioned in Subsection 3.1 is linear integer programming, and a Lagrangian relaxation method can be used to dualize these equality constraints, the multi-equations-based unique path constraint (9) consists of a large number of $\frac{|V| \times (D-1) \times D}{2}$ equality relationships, where $|V|$ is the cardinality of the set $V$ of physical links. To capture the essential modeling requirement in a more convenient way, we begin to examine its equivalent forms.

**Lemma 1** The multi-equations-based unique path constraint (9) is equivalent to

$$
\sum_{d=1}^{D} \sum_{(i, j) \in V} \left( x^d_{ij} - \bar{x}_j \right)^2 = 0,
$$

$$
\bar{x}_j = \frac{1}{D} \times \left( \sum_{d=1}^{D} x^d_{ij} \right), (i, j) \in V. \hspace{1cm} (12)
$$

**Proof.** If the unique path constraint (9) holds, then the set of links on the physical network can be divided into two subsets, denoted by $E$ and $F$, respectively, satisfying:

(i) $x^d_{ij} = 0$ for all $(i, j) \in E, \ d = 1, 2, \ldots, D$ for links not being selected in the LET solution;

(ii) $x^d_{ij} = 1$ for all $(i, j) \in F, \ d = 1, 2, \ldots, D$ for links being selected in the LET solution.
From the definitional constraint (12), case (i) yields \( x^d_{ij} - \overline{x}_{ij} = 0 \) for any link \((i, j) \in E\), \(d = 1, 2, \cdots, D\), resulting in \( \sum_{d=1}^{D} \sum_{(i, j) \in E} (x^d_{ij} - \overline{x}_{ij})^2 = 0 \) as a single equation across all links. In case (ii), it is easy to verify that \( \overline{x}_{ij} = 1 \) for any link \((i, j) \in F\), which leads to \( x^d_{ij} - \overline{x}_{ij} = 0 \) for each link \((i, j) \in F\), \(d = 1, 2, \cdots, D\). In other words, we have \( \sum_{d=1}^{D} \sum_{(i, j) \in F} (x^d_{ij} - \overline{x}_{ij})^2 = 0 \). Alternatively, constraint (11) can be decomposed as
\[
\sum_{d=1}^{D} \sum_{(i, j) \in V} (x^d_{ij} - \overline{x}_{ij})^2 = \sum_{d=1}^{D} \sum_{(i, j) \in E} (x^d_{ij} - \overline{x}_{ij})^2 + \sum_{d=1}^{D} \sum_{(i, j) \in F} (x^d_{ij} - \overline{x}_{ij})^2 = 0.
\]
If equation (11) is true, we have \( x^d_{ij} - \overline{x}_{ij} = 0 \) for all \((i, j) \in E\), \(d = 1, 2, \cdots, D\), resulting in
\[
x^{d=1} = x^{d=2} = \cdots = x^{d=D}, (i, j) \in V. \]

Obviously, a large number of equalities in constraint (9) can be dramatically simplified by Lemma 1 to a small constraint set with a size of \(|V| + 1\) definitional constraints associated with \( x^d_{ij} \) and \( \overline{x}_{ij} \).

In real-world large-scale network applications, constraint (11) may remain difficult to solve because of its form of equality restriction. A further relaxed alternative constraint, referred to as a quadratic variance-based unique path constraint, is introduced in this study as the second model reformulation. A positive threshold value \( \frac{1}{D^2} \) is used to set up the following inequality constraint, where \( D \) is the total number of scenarios.
\[
\sum_{d=1}^{D} \sum_{(i, j) \in V} (x^d_{ij} - \overline{x}_{ij})^2 < \frac{1}{D^2}. \tag{13}
\]

**Lemma 2** If \( X \in \overline{B} \), constraint (11) is equivalent to constraint (13). That is,
\[
\sum_{d=1}^{D} \sum_{(i, j) \in V} (x^d_{ij} - \overline{x}_{ij})^2 = 0 \iff \sum_{d=1}^{D} \sum_{(i, j) \in V} (x^d_{ij} - \overline{x}_{ij})^2 < \frac{1}{D^2}.
\]

**Proof.** Constraint (11) yields constraint (13) because the former can be viewed as a special case with a smaller feasible region. Next, we need to prove that constraint (13) leads to constraint (11) by contradiction.

If there exist two days \( d0 \) and \( d1 \) with different link selection indicators such that \( x^{d0}_{ij} \neq x^{d1}_{ij} \) for a given link \((i, j) \in V\), from the definition of \( \overline{x}_{ij} \), we can derive \( \overline{x}_{ij} \notin \{0,1\} \), namely, \( \overline{x}_{ij} \in \left\{ \frac{1}{D}, \frac{2}{D}, \cdots, \frac{D-1}{D} \right\} \). Without the loss of generality, we consider two particular days \( x^{d1}_{ij} = 1 \) and \( x^{d0}_{ij} = 0 \) among all \( D \) days. Then, it follows that \( (x^{d0}_{ij} - \overline{x}_{ij})^2 = \overline{x}_{ij}^2 \geq \frac{1}{D^2} \) and \( (x^{d1}_{ij} - \overline{x}_{ij})^2 = (1-\overline{x}_{ij})^2 \geq \frac{1}{D^2} \), so the sum over all \( D \) days is \( \sum_{d=4}^{D} \sum_{(i, j) \in V} (x^d_{ij} - \overline{x}_{ij})^2 > \frac{1}{D^2} \). In other words, if condition \( \sum_{d=1}^{D} \sum_{(i, j) \in V} (x^d_{ij} - \overline{x}_{ij})^2 < \frac{1}{D^2} \) holds, then there are no two days with different selection indicators; that is, \( x^{d=1}_{ij} = x^{d=2} = \cdots = x^{d=D}, (i, j) \in V \), satisfying equation (11). \n
In the following discussion, the formulation \( \sum_{d=1}^{D} \sum_{(i, j) \in V} (x^d_{ij} - \overline{x}_{ij})^2 \) will be abbreviated by a vector-based
representation \( \|X - \bar{X}\|^2 \), and the definitional constraint (12) for the mean link selection rate will be represented by \( \bar{X} = I(X) \), where \( I(\cdot) \) denotes a linear transformation.

### 3.3. Constructing Equivalent Linear Inequality

The unique a priori path constraint is now reformulated and expressed through equations (11) and (13). Because the standard shortest path algorithm accepts only linear formulations, the nonlinear function constraints (11) and (13) might not be desirable. To avoid complex linearization-related computation, this study offers a linear equivalent as below.

**Lemma 3** If \( X \in \bar{B} \), equalities \( x_{ij}^{d+1} = x_{ij}^{d+2} = \cdots = x_{ij}^{d+D} \), \( \forall (i, j) \in V \) hold if and only if

\[
x_{ij}^d - \bar{x}_{ij} < \frac{1}{D}, \forall (i, j) \in V, d = 1, 2, \cdots, D.
\]  

**Proof.** Because the necessary condition is obvious, we need to prove the sufficient condition by contradiction. Consider that equation (14) holds, but the unique path constraint is not true. Then, there must exist a link \( (i, j) \in V \) and two of the scenarios \( d0 \) and \( d1 \) from days \( d = 1, 2, \cdots, D \) such that \( x_{ij}^{d0} \neq x_{ij}^{d1} \), which implies that one of two variables \( x_{ij}^{d0} \) and \( x_{ij}^{d1} \) has a value of 0, \( \bar{x}_{ij} \) is strictly less than 1 and \( \bar{x}_{ij} \leq \frac{D-1}{D} \). Without the loss of generality, we set \( x_{ij}^{d1} = 1 \) and \( x_{ij}^{d0} = 0 \). Then, it is easy to obtain \( x_{ij}^{d1} - \bar{x}_{ij} \geq \frac{1}{D} \). Thus, a contradiction proves the sufficient condition. ■

Compared with constraint (9), equations (14) and (12) are also simplified formulations because the total number of constraints in (14) and (12) is \( |V| \times (D + 1) \), and both sets of constraints have linear forms. To further simplify the notation, we introduce a smaller positive threshold value \( \nu = \frac{1}{D} - \epsilon \), where \( 0 < \epsilon < \frac{1}{D} \), and denote constraint (14) by “\( X - \bar{X} \leq \nu \)”. Then, the fourth version of the original problem is formulated by

\[
\begin{align*}
\min & \quad \text{expected total travel time (5)} \\
\text{s.t.} & \quad F(X, Y) = CY \\
& \quad AX^d = b, \ d = 1, 2, \cdots, D \\
& \quad X - \bar{X} \leq \nu \\
& \quad \bar{X} = I(X) \\
& \quad B^d Y^d = h^d, \ \forall \ d = 1, 2, \cdots, D \\
& \quad Y^d \leq X^d, \ d = 1, 2, \cdots, D \\
& \quad X, Y \in \bar{B}
\end{align*}
\]  

### 4. Lagrangian Relaxation

To dualize hard constraints and search for tight lower bounds of model (15), this section will present a Lagrangian relaxation approach for solving the least expected time shortest path problem.

#### 4.1. Dualizing Hard Constraints.

In model (15), constraints (14) and (12) are considered complex constraints to be dualized. In particular, an auxiliary variable \( \bar{x}_{ij} \) will be introduced for the definitional constraint (12). Two sets of multipliers are also needed for dualizing constraints (14) and (12), i.e., unique path constraint multiplier \( \beta_{ij}^d \geq 0 \) and mean
link selection rate constraint multiplier \( \gamma_{ij} \in \mathbb{R} \), where \((i, j) \in \mathcal{V}, d = 1, 2, \cdots, D \). Although the domain of \( \bar{x}_{ij} \) is a finite discrete set of \( \left\{ 0, 1, \frac{2}{D}, \cdots, 1 \right\} \), we herein still relax the value of \( \bar{x}_{ij} \) in a continuous interval \([0, 1]\), denoted by \( \bar{X} \in [0,1] \) for simplicity. At this point, model (14) can be reformulated as:

\[
\begin{align*}
\min & \sum_{d=1}^{D} \sum_{(i,j) \in \mathcal{V}} \sum_{t \in \mathcal{T}} c_{ij}^{d} y_{ij}^{d} + \sum_{d=1}^{D} \sum_{(i,j) \in \mathcal{V}} \beta_{ij}^{d} \left( x_{ij}^{d} - \bar{x}_{ij} - \gamma \right) + \sum_{(i,j) \in \mathcal{V}} \gamma_{ij} \left( \frac{1}{D} \times \left( \sum_{d=1}^{D} x_{ij}^{d} \right) - \bar{x}_{ij} \right) \\
\text{s.t.} & \quad \{\text{physical network flow balance constraint (8)}\} \quad AX^{d} = b, \; d = 1, 2, \cdots, D \\
& \quad \{\text{space-time flow balance constraint (2)}\} \quad B^{d}X^{d} = h^{d}, \; d = 1, 2, \cdots, D \\
& \quad \{\text{space-time arc-to-link mapping constraint (10)}\} \quad Y^{d} \leq X^{d}, \; d = 1, 2, \cdots, D \\
& \quad \{\text{binary variable constraint (4)}\} \quad X, Y \in \bar{B} \\
& \quad \{\text{auxiliary variable constraint}\} \quad \bar{X} \in [0,1].
\end{align*}
\]

By regrouping the variables, we finally have a more systematic view of the components of the dual problem:

\[
\begin{align*}
\min & \sum_{d=1}^{D} \sum_{(i,j) \in \mathcal{V}} \sum_{t \in \mathcal{T}} c_{ij}^{d} y_{ij}^{d} + \sum_{d=1}^{D} \sum_{(i,j) \in \mathcal{V}} \left( \beta_{ij}^{d} + \frac{\gamma_{ij}}{D} \right) x_{ij}^{d} - \sum_{d=1}^{D} \sum_{(i,j) \in \mathcal{V}} \left( \beta_{ij}^{d} + \frac{\gamma_{ij}}{D} \right) \bar{x}_{ij} - \gamma \sum_{d=1}^{D} \sum_{(i,j) \in \mathcal{V}} \beta_{ij}^{d} \\
\text{s.t.} & \quad \{\text{physical network flow balance constraint (8)}\} \quad AX^{d} = b, \; d = 1, 2, \cdots, D \\
& \quad \{\text{space-time flow balance constraint (2)}\} \quad B^{d}X^{d} = h^{d}, \; d = 1, 2, \cdots, D \\
& \quad \{\text{space-time arc-to-link mapping constraint (10)}\} \quad Y^{d} \leq X^{d}, \; d = 1, 2, \cdots, D \\
& \quad \{\text{binary variable constraint (4)}\} \quad X, Y \in \bar{B} \\
& \quad \{\text{auxiliary variable constraint}\} \quad \bar{X} \in [0,1].
\end{align*}
\]

4.2. Problem Decomposition

It is easy to prove that the optimal objective of model (16) is a lower bound of the optimal objective to the primal model. To solve model (16) for any given Lagrangian multiplier vector \((\beta, \gamma)\), we decompose this model into two sub-problems.

- Sub-problem 1 \( SPI(\beta, \gamma) \): time-dependent shortest path problem with generalized costs

\[
\begin{align*}
\min & \sum_{d=1}^{D} \sum_{(i,j) \in \mathcal{V}} \sum_{t \in \mathcal{T}} c_{ij}^{d} y_{ij}^{d} + \sum_{d=1}^{D} \sum_{(i,j) \in \mathcal{V}} \left( \beta_{ij}^{d} + \frac{\gamma_{ij}}{D} \right) x_{ij}^{d} \\
\text{s.t.} & \quad \{\text{physical network flow balance constraint (8)}\} \quad AX^{d} = b, \; d = 1, 2, \cdots, D \\
& \quad \{\text{space-time flow balance constraint (2)}\} \quad B^{d}X^{d} = h^{d}, \; d = 1, 2, \cdots, D \\
& \quad \{\text{space-time arc-to-link mapping constraint (10)}\} \quad Y^{d} \leq X^{d}, \; d = 1, 2, \cdots, D \\
& \quad \{\text{binary variable constraint (4)}\} \quad X, Y \in \bar{B}.
\end{align*}
\]

Model (17) involves decision variables \( y_{ij}^{d} \) and \( x_{ij}^{d} \), and it can be further separated into \( D \) sub-problems, denoted by \( SPI(\beta, \gamma, d) \), \( d = 1, 2, \cdots, D \), which are referred to as the scenario-based time-dependent shortest path problems with generalized costs. That is,

\[
\begin{align*}
\min & \sum_{(i,j) \in \mathcal{V}} \sum_{t \in \mathcal{T}} c_{ij}^{d} y_{ij}^{d} + \sum_{(i,j) \in \mathcal{V}} \left( \beta_{ij}^{d} + \frac{\gamma_{ij}}{D} \right) x_{ij}^{d} \\
\text{s.t.} & \quad \{\text{physical network flow balance constraint (8)}\} \quad AX^{d} = b, \; d = 1, 2, \cdots, D \\
& \quad \{\text{space-time flow balance constraint (2)}\} \quad B^{d}X^{d} = h^{d}, \; d = 1, 2, \cdots, D \\
& \quad \{\text{space-time arc-to-link mapping constraint (10)}\} \quad Y^{d} \leq X^{d}, \; d = 1, 2, \cdots, D \\
& \quad \{\text{binary variable constraint (4)}\} \quad X, Y \in \bar{B}.
\end{align*}
\]
s.t. \( AX^d = b \) and \( B^d Y^d = h^d \) and \( Y^d \leq X^d \) and \( X^d, Y^d \in \mathcal{B} \).

In fact, for each scenario \( d \in \{1, 2, \cdots, D\} \), the sub-problem \( SPI(\beta, \gamma, d) \) can be solved by efficient label-setting or label-correcting algorithms. The optimal objective of model (18) will be denoted by \( Z^*_{SPI}(\beta, \gamma, d) \).

Based on Bellman’s principle of optimality, modified label-correcting algorithms (Ziliaskopoulos and Mahmassani 1992; Pallottino and Scutella 1997) can be adopted to search for the optimal time-dependent least-cost paths of model (18). In this iterative algorithm, the generalized link cost consists of two parts: (1) the time-dependent link travel time \( c^i_{jt} \), whose value is dependent on both time stamps \( t \in T \) and physical links \( (i, j) \in V \), and (2) the penalty and time-invariant cost \( \beta^d_j + \frac{\gamma^d_j}{D} \) associated with the selected physical links, which are correlated to each physical link \((i, j) \in V \). Then, the generalized time-dependent link cost, denoted by \( g^d_{ij} \), can be calculated by

\[
g^d_{ij} = c^i_{jt} + \beta^d_j + \frac{\gamma^d_j}{D}.
\]  

(19)

In the searching procedure, an \( M \)-dimensional vector-based label, denoted by \( \Gamma_j = (\lambda_1(t_0), \lambda_2(t_0 + \sigma), \cdots, \lambda_M(t_0 + M\sigma)) \), can be used to show the least travel cost from origin \( O \) to current node \( j \) at each time stamp \( t' \in T \), where \( \lambda_j(t') \) is computed by \((t' = t + c^d_{ij})\):

\[
\lambda_j(t') = \min \left\{ \lambda_j(t'), g^d_{ij} + \lambda_i(t) \right\}, \quad j \in V \setminus i, \quad t' \in T.
\]  

(20)

After solving sub-problems \( SPI(\beta, \gamma, d), d = 1, 2, \cdots, D \), the sum of corresponding least travel costs (across different scenarios) becomes the optimal objective of model \( SPI(\beta, \gamma) \).

• Sub-problem 2 \( SP2(\beta, \gamma) \): auxiliary-variable optimization problem

\[
SP2(\beta, \gamma) : \max \left\{ \sum_{(i,j) \in V} \left( \sum_{d=1}^{D} \beta^d_j + \gamma^d_j \right) \bar{x}_{ij} : \bar{x} \in [0,1] \right\}.
\]  

(21)

Model \( (SP2(\beta, \gamma)) \) can be further re-expressed in terms of the following link-based auxiliary-variable optimization problem for each link \((i, j) \in V \), namely,

\[
SP2(\beta, \gamma, i, j) : \max \left\{ \left( \sum_{d=1}^{D} \beta^d_j + \gamma^d_j \right) \bar{x}_{ij} : \bar{x}_{ij} \in [0,1] \right\}.
\]  

(22)

Equation (22) is a simple univariable linear program with a bound, in which unique path constraint multiplier \( \beta^d_j \) and definitional mean link selection rate constraint multiplier \( \gamma^d_j \) are input parameters predetermined in model (16). For each model \( SP2(\beta, \gamma, i, j) \), we can calculate its optimal objective as follows:

\[
Z^*_{SP2}(\beta, \gamma, i, j) = \left\{ \begin{array}{ll}
\sum_{d=1}^{D} \beta^d_j + \gamma^d_j, & \text{if } \sum_{d=1}^{D} \beta^d_j + \gamma^d_j \geq 0, \\
0, & \text{otherwise}
\end{array} \right.
\]  

(23)

After solving \( SP2(\beta, \gamma, i, j) \) for each link \((i, j) \in V \), the sum of corresponding optimal objective values equals the optimal objective value of model \( SP2(\beta, \gamma) \).

For any Lagrangian multipliers \((\beta, \gamma)\), the optimal objective value of model (16), denoted by \( LR(\beta, \gamma) \), can be computed by the formulation below:
\[ LR(\beta, \gamma) = \sum_{d=1}^{D} Z_{1}^{SP1}(\beta, \gamma, d) - \sum_{(i,j) \in V} \sum_{(i,j) \in V} Z_{1}^{SP2}(\beta, \gamma, i, j) - v \sum_{d=1}^{D} \sum_{(i,j) \in V} \beta_{ij}^{d}. \]  

(24)

4.3. Comparison with a Wait and See Solution

After we present the Lagrangian relaxation model, it would be interesting to compare the corresponding tightest lower bound with the well-known *Wait and See* (WAS) bound for stochastic programming, which aims to choose the optimal paths based on the realized link travel time of each scenario. In detail, for each scenario \( d \), we first search for a time-dependent shortest path associated with the realized sample data. The summation, i.e., \( \sum_{d=1}^{D} T[X_d] \), offers a lower bound of the optimal objective value to the primal stochastic programming problem, where \( T(X_d) \) denotes the least travel time on scenario \( d \).

**Lemma 4** If \( \beta_{ij}^{d} = 0, \gamma_{ij} = 0 \) for any \( (i,j) \in V \), \( d = 1, 2, \ldots, D \), then the Lagrangian relaxation-based model (16) is degenerated to a WAS model.

That is, compared with our proposed Lagrangian relaxation-based approach, a WAS solution does not impose the unique a priori path constraint, which leads to a larger feasible region and then produces a relatively loose lower bound compared with the original model (15).

5. Solution Algorithm

5.1. Solution Strategies

Because the primal model is to minimize the total travel time over different scenario-based data, the path travel time of each feasible path solution in model (15) can be regarded as an upper bound of the optimal objective to the original problem. To find an approximate optimal solution with the guaranteed quality, the proposed iterative solution algorithm is intended to minimize the gap between the upper bound and the lower bound. Particularly, if the minimized gap is equal to 0, the exact optimal solution will be obtained.

In this section, a sub-gradient algorithm will be designed to improve the lower bound. This solution technique aims to decrease the dual gap iteratively by updating the Lagrangian multipliers \((\beta, \gamma)\) along the sub-gradient direction of the objective function in model (16). For fixed values of multipliers \((\beta, \gamma)\), the optimal objective \( LR(\beta, \gamma) \) of model (16) is a lower bound of the optimal objective of the primal model (15), and the following Lagrangian dual problem needs to be solved:

\[ Z_{LR}^{*} = LR(\beta^*, \gamma^*) = \max_{\beta,\gamma} \{ LR(\beta, \gamma) \}. \]  

(25)

Meanwhile, the upper bound will be updated using newly available better feasible path solutions (with a lower objective function (5)) to minimize the dual gap.

To iteratively update the parameter vector \((\beta, \gamma)\) for reaching tighter lower bounds, at each iteration, the vector components of sub-gradient will be calculated by:

\[ \nabla LR_{\beta} (\beta, \gamma) = x_{ij} - \bar{x}_{ij} - v, \quad \nabla LR_{\gamma} (\beta, \gamma) = \frac{1}{D} \times \left( \sum_{d=1}^{D} x_{ij}^{d} \right) - \bar{x}_{ij}. \]  

(26)

for any \((i,j) \in V, d = 1, 2, \ldots, D\). Consequently, the sub-gradient vector \( \nabla LR(\beta, \gamma) \), consisting of \( \nabla LR_{\beta} (\beta, \gamma) \) and \( \nabla LR_{\gamma} (\beta, \gamma) \), \( (i,j) \in V, d = 1, 2, \ldots, D \), will be treated as the searching direction for the next iteration.

In the searching process, the algorithm starts with a predetermined initial solution, and notation \( k \) is used as the index of the iterative number. Denote the Lagrangian multiplier vector by \( (\hat{\beta}_{ij}, \hat{\gamma}_{ij}) \) at iteration \( k \). To obtain the updated multipliers for the next iteration, first we need to solve model (16) at the current iteration, where variables of the optimal solution are denoted by \( \hat{y}_{ij}^{d,k}, \hat{x}_{ij}^{d,k} \) and \( \bar{x}_{ij}^{*} \), \( \forall (i,j) \in V, t \in T, d = 1, 2, \ldots, D \), respectively. Then, the following equations will be used to update the Lagrangian multipliers for iteration \( k+1 \):

\[
\begin{align*}
    &\hat{\beta}_{ij}^{k+1} = \hat{y}_{ij}^{d,k} + \bar{x}_{ij}^{*} + v, \\
    &\hat{\gamma}_{ij}^{k+1} = \frac{1}{D} \times \left( \sum_{d=1}^{D} \hat{x}_{ij}^{d,k} \right) - \bar{x}_{ij}^{*}.
\end{align*}
\]
\[
\beta^{d,k+1}_j = \beta^{d,k}_j + \theta_k (x^{d,k}_j - \bar{x}^k_j - v), \quad \gamma^{k+1}_j = \gamma^k_j + \theta_k \left( \frac{1}{D} \sum_{d=1}^{D} x^{d,k}_j - \bar{x}^k_j \right)
\]

for any \((i, j) \in V, d = 1, 2, \cdots, D\), where the parameter \(\theta_k\) is determined as follows:

\[
\theta_k = \frac{\lambda_k (UB_k - LR(\beta_k, \gamma_k))}{f(x^{d,k}_j, \bar{x}^k_j)},
\]

\[
f(x^{d,k}_j, \bar{x}^k_j) = \left( \sum_{d=1}^{D} \sum_{d(\ell, j)} (x^{d,k}_j - \bar{x}^k_j - v)^2 + \sum_{d(\ell, j)} \left( \frac{1}{D} \sum_{d=1}^{D} x^{d,k}_j - \bar{x}^k_j \right)^2 \right)^{\frac{1}{2}}.
\]

In equation (28), notation \(UB_k\) represents the best objective value encountered so far in the primal problem and can be updated iteratively to speed up the optimization process. The parameter \(\lambda_k\) is a scalar in interval \([0, 2]\), the purpose of which is to adjust the step size of the process and guarantee that no negative cost appears in the objective function.

In the proposed algorithm, \(UB_k\) can be iteratively updated by solving sub-problems \(SP(\beta_k, \gamma_k, d)\), \(d \in \{1, 2, \cdots, D\}\) at each iteration \(k\). Specifically, let \(UB^d_k\) denote the objective of model (15) corresponding to the optimal routing plan of \(SP(\beta_k, \gamma_k, d)\); then, the upper bound at iteration \(k\) will be updated by the following formulation:

\[
UB_k \leftarrow \min \left\{ UB_{k-1}, \min_{d \in \{1, 2, \cdots, D\}} UB^d_k \right\}.
\]

Once the dual gap \(UB_k - LR(\beta_k, \gamma_k)\) becomes less than a predetermined toleration gap \(\epsilon\), or the iteration \(k\) is larger than a predefined maximum iteration \(K_{\text{max}}\), the algorithm will terminate, and the \(UB_k\) will be used to evaluate the solution quality in terms of gaps.

### 5.2. Solution Procedure

We now present the complete solution procedure as follows:

**Step 1. (Initialization)** Let \(k = 1\) (referred to as inner loop iteration index in numerical experiments); initialize the multipliers \((\beta_k, \gamma_k)\), set \(UB_k\) as a sufficiently number or an objective value of a feasible solution to the primal model.

**Step 2. (Solve the Relaxed Model)** This part is divided into two types of sub-problems, where parameters \((\beta_k, \gamma_k)\) are used.

1. Solve the model (18) for each \(d = 1, 2, \cdots, D\) by a modified label-correcting algorithm;
2. Compute the equality (23) for each \((i, j) \in V\).

**Step 3. (Compute the Gap)** Denote the optimal solution of the above problem by \(\gamma^{d,k}_{ij}, x^{d,k}_{ij}\) and \(\bar{x}^k_j\), \(\forall (i, j) \in V, d = 1, 2, \cdots, D, t \in T\). Based on the optimal solutions, calculate primal, dual and gap values.

**Step 4. (Update Lagrangian Multipliers)** Compute the multipliers for the next iteration by equation (27).

**Step 5. (Termination Conditions)** If \(k > K_{\text{max}}\) (a predetermined maximum number of iterations) or the gap is less than the predetermined value \(\epsilon\), then the algorithm will be terminated. Otherwise, let \(k \leftarrow k + 1\), go to Step 2.

To clarify the procedure of the solution methodologies, a flow chart illustrating each step of the Lagrangian relaxation approach is shown in Figure 4.
6. Numerical Experiments

In this section, the effectiveness and computational efficiency of the Lagrangian relaxation method is tested by using a simplified network and two real-world networks. The algorithm is implemented in C++ on the Windows 7.0 platform and evaluated on a personal computer with an Intel(R) Core(TM) i5 CPU and 4 GB memory.

6.1. Illustration of the Solution Approach with a Simplified Network

In the first set of numerical experiments, we consider a simple transportation network, as shown in Figure 1, in which the travel time on each link is assumed to be time-invariant, and the travel time information is given in Table 6 with two scenarios \( d = 1, 2 \). For this special case, decision variables in model (15) will be degenerated to a single decision variable \( x_{dj}^d \), and time-dependent link travel time \( c_{ij}^d \) will be reduced to a time-invariant parameter \( c_{ij} \).

Table 6 Travel Time on Each Link for Different Scenarios (Unit: min)

<table>
<thead>
<tr>
<th>Scenarios</th>
<th>Link (1,2)</th>
<th>Link (1,3)</th>
<th>Link (2,3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>( d = 1 )</td>
<td>6</td>
<td>7</td>
<td>3</td>
</tr>
<tr>
<td>( d = 2 )</td>
<td>3</td>
<td>8</td>
<td>2</td>
</tr>
</tbody>
</table>

First, we apply the optimization model (15) with an OD pair from node 1 to node 3, where \( v \) is set as a small value \( 0.001 < \frac{1}{2} \) and where 2 indicates the number of scenarios. Because there are only 2 paths between this OD pair, by enumeration, one can simply obtain the optimal solution \( 1 \rightarrow 2 \rightarrow 3 \) with an objective function \( Z_{\text{LET}}^* = 14 \), where \( Z_{\text{LET}}^* \) is the optimal objective value of the primal model. As analyzed in Subsection 4.3, the WAS model in this case leads to the two shortest paths corresponding to different scenarios; that is, path \( 1 \rightarrow 3 \) on scenario 1 and path \( 1 \rightarrow 2 \rightarrow 3 \) on scenario 2, which leads to a WAS-based lower bound of 12 min. The detailed results are shown in Table 7.

Table 7 Solution Provided by Wait and See Model

<table>
<thead>
<tr>
<th>Scenarios</th>
<th>( x_{12}^d )</th>
<th>( x_{13}^d )</th>
<th>( x_{23}^d )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( d = 1 )</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
As expected, $12 = Z_{WAS}^* < Z_{LR}^* = 14$. In comparison, solving the Lagrangian relaxation model (16) in Subsection 4.1 yields computational results in the first 10 iterations listed in Table 8.

<table>
<thead>
<tr>
<th>Iteration</th>
<th>$\beta_{12}^{d=1}$</th>
<th>$\beta_{13}^{d=1}$</th>
<th>$\beta_{23}^{d=1}$</th>
<th>$\beta_{12}^{d=2}$</th>
<th>$\beta_{13}^{d=2}$</th>
<th>$\beta_{23}^{d=2}$</th>
<th>$\gamma_{12}$</th>
<th>$\gamma_{13}$</th>
<th>$\gamma_{23}$</th>
<th>$LB$</th>
<th>$UB$</th>
<th>Gap</th>
<th>Relative Gap</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>5.65</td>
<td>6.48</td>
<td>6.03</td>
<td>7.26</td>
<td>6.52</td>
<td>6.61</td>
<td>-11.65</td>
<td>-10.85</td>
<td>-10.70</td>
<td>10.35</td>
<td>14.00</td>
<td>3.65</td>
<td>26.05%</td>
</tr>
<tr>
<td>4</td>
<td>2.41</td>
<td>6.48</td>
<td>2.80</td>
<td>7.26</td>
<td>3.29</td>
<td>6.60</td>
<td>-13.26</td>
<td>-12.46</td>
<td>-12.31</td>
<td>10.35</td>
<td>14.00</td>
<td>3.65</td>
<td>26.05%</td>
</tr>
<tr>
<td>5</td>
<td>7.95</td>
<td>6.47</td>
<td>8.34</td>
<td>7.25</td>
<td>8.83</td>
<td>6.60</td>
<td>-10.49</td>
<td>-9.69</td>
<td>-9.54</td>
<td>10.35</td>
<td>14.00</td>
<td>3.65</td>
<td>26.05%</td>
</tr>
<tr>
<td>6</td>
<td>1.43</td>
<td>6.47</td>
<td>1.81</td>
<td>7.25</td>
<td>2.30</td>
<td>6.60</td>
<td>-13.75</td>
<td>-12.95</td>
<td>-12.80</td>
<td>10.35</td>
<td>14.00</td>
<td>3.65</td>
<td>26.05%</td>
</tr>
<tr>
<td>7</td>
<td>4.95</td>
<td>6.46</td>
<td>5.34</td>
<td>7.24</td>
<td>5.83</td>
<td>6.59</td>
<td>-11.98</td>
<td>-11.18</td>
<td>-11.04</td>
<td>12.86</td>
<td>14.00</td>
<td>1.14</td>
<td>8.16%</td>
</tr>
<tr>
<td>8</td>
<td>4.95</td>
<td>6.44</td>
<td>5.34</td>
<td>7.24</td>
<td>5.81</td>
<td>6.59</td>
<td>-11.98</td>
<td>-11.20</td>
<td>-11.04</td>
<td>12.92</td>
<td>14.00</td>
<td>1.08</td>
<td>7.74%</td>
</tr>
<tr>
<td>9</td>
<td>4.95</td>
<td>6.33</td>
<td>5.34</td>
<td>7.24</td>
<td>5.70</td>
<td>6.59</td>
<td>-11.98</td>
<td>-11.32</td>
<td>-11.04</td>
<td>13.15</td>
<td>14.00</td>
<td>0.85</td>
<td>6.10%</td>
</tr>
<tr>
<td>10</td>
<td>4.63</td>
<td>6.33</td>
<td>5.02</td>
<td>7.24</td>
<td>5.37</td>
<td>6.59</td>
<td>-12.14</td>
<td>-11.48</td>
<td>-11.20</td>
<td>13.48</td>
<td>14.00</td>
<td>0.52</td>
<td>3.74%</td>
</tr>
</tbody>
</table>

In Table 8, $LB$ and $UB$, respectively, denote the best lower bound and upper bound encountered up to the current iteration, whereas Gap and Relative Gap are calculated by the formulations below:

$$\text{Gap} = UB - LB,$$

$$\text{Relative Gap} = \frac{UB - LB}{UB} \times 100\%.$$  

A lower bound of 13.48 is obtained at iteration 10, which shows only that the proposed Lagrangian relaxation-based approach can provide a much tighter lower bound than the WAS method, as $Z_{WAS}^* = 12 < Z_{LR}^* = 13.48 \leq Z_{LET}^* = 14$.

For this simple network, a total of 9 multipliers need to be considered in the proposed Lagrangian dual problem, including 6 multipliers for unique path constraints and 3 multipliers associated with equality constraints for calculating the mean of link selection indicators across different scenarios. The close-to-optimal values for multipliers $\beta_{13}^{d=1}, \beta_{12}^{d=2}$ and $\beta_{23}^{d=2}$ (shown as shaded in Table 8) are quickly discovered at the third iteration, and their values remain nearly unchanged afterward. The searching process continues to adjust the values of the other Lagrangian multipliers, and the relative solution quality gap gradually reduces to 3.74%, as shown in Figure 5. It is noteworthy that before iteration 10, the lower bound solution still generates two different paths, even with a relatively high lower bound value. After iteration 10, the scenario-dependent solutions will become reduced to a single unique path.
6.2. Numerical Experiments for a Medium-Scale Network

The second set of experiments considers a real-world freeway network with 127 nodes and 284 links extracted from the San Diego region (see Figure 6, left: Google Maps, right: node-link network structure).

This experiment considers a one-hour planning time horizon with a 0.5-min travel time aggregation time interval. Ten days (i.e., scenarios) of time-dependent travel time data are randomly simulated by using a data-generation method similar to that proposed by Miller-Hooks (1997). Specifically, the free-flow travel time of each link \((i, j)\), denoted by \(\bar{t}_{ij}\), is first calculated. Then, we generate a random integer \(q^d_t\) in set \{1, 2\} for time interval \(t\) to construct \(q^d_t \cdot \bar{t}_{ij}\) as the link travel time on link \((i, j)\) at departure time \(t\) on scenario \(d\). Considering the OD pair shown in Figure 6, a total of 3124 Lagrangian multipliers need to be considered in the proposed Lagrangian dual model. Thus, to avoid generating only local optimums, a global optimization searching method is used in the experiments, in which multiple initial solutions are produced to restart the proposed sub-gradient algorithm for every 15 inner iterations (i.e., \(K_{\text{max}} = 15\)).

After executing the algorithm for 20 (randomly generated) starting seeds of Lagrangian multipliers, the obtained best upper bound and tightest lower bound of the primal problem are \(Z_{\text{LET}}^* = 254.11\) and \(Z_{\text{LR}}^* = 247.57\), where Relative Gap = 2.57%. In comparison, the WAS method generated a lower bound of \(Z_{\text{WAS}}^* = 236.27\) with a relative gap of 7.02% (see Table 9). The computational results show that the proposed Lagrangian relaxation approach can find a high-quality solution and a much tighter lower bound for the primal problem.
Table 9 Optimal Solution and Lower Bounds

<table>
<thead>
<tr>
<th></th>
<th>( Z_{LET} )</th>
<th>( Z_{LR} )</th>
<th>( Z_{WAS} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value</td>
<td>254.11</td>
<td><strong>247.57</strong></td>
<td>236.27</td>
</tr>
<tr>
<td>Gap</td>
<td>(--)</td>
<td>6.54</td>
<td>17.84</td>
</tr>
<tr>
<td>Relative Gap</td>
<td>(--)</td>
<td>2.57%</td>
<td>7.02%</td>
</tr>
</tbody>
</table>

To show the searching process clearly, Figure 7 displays the variation of the lower bounds with respect to different seeds of initial multipliers. The x-axis represents the iteration indices for generating initial multipliers in the searching process (i.e., outer loop iteration index). With different initial solutions, the lower bound encountered so far (Line “LB(OI),” Figure 7) will be updated in a non-decreasing sequence. When the outer loop iteration index equals 2, the tightest lower bound \( Z_{LR} \) is obtained with the smallest relative gap. Figure 8 shows the variation of relative gaps with the increase of the outer loop iteration.

Clearly, in each iteration of the searching process, the most time-consuming step of the sub-gradient algorithm is the time-dependent shortest path generation for different scenarios. Thus, the computational complexity of the proposed algorithm is dependent to a great extent on the scale of the network and the number of scenarios. In the experiments, for the computational process of the sub-gradient algorithm with a maximum of 15 inner iterations, the general time consumption in each iteration is about 4 to 7 seconds (including 10 time-dependent shortest path solving steps).

**Figure 7 Variations of Lower Bounds and Upper Bounds**

- **LB(II):** The best lower bound obtained with respect to different restarting seeds of initial multipliers;
- **LB(OI):** The tightest lower bound obtained up to outer loop iteration \( l \);
- **UB:** The variation of upper bounds;
- **WAS:** The variation of lower bounds by the WAS.
Figure 8: Variation of the Relative Gap Associated with Different Sets of Initial Lagrangian Multipliers

RelGap1: The best relative gap associated with different restarting seeds of initial multipliers;
RelGap2: The best relative gap obtained up to outer loop iteration $l$.

6.3. Numerical Experiments for a Large-scale Network with Real-world Sensor Data

The third set of numerical experiments is performed on a part of the real-world transportation network near Den Haag-Rotterdam in the Netherlands with the recorded time-dependent link travel time data obtained by sensors. As shown in Figure 9 (left: Google Maps, right: node-link network structure), an abstract network, consisting of 299 nodes and 736 links, is extracted from the real network for the experiments; links marked by squares are equipped with sensors for obtaining the real-time link travel times.
In the experiments, a set of ten days of time-dependent link travel times obtained by 130 point detectors are used to search for the robust a priori LET routes. Specifically, the time horizon is considered to be from 15:30 to 17:00 and is discretized by 0.1-min time intervals. For the links equipped with the sensors, the real-time dependent travel time data are collected from 15:30 to 17:00 on ten weekdays. Meanwhile, for the links without detectors, the region-wide link travel time index is adopted as the default values. Additionally, to simulate the impact of non-recurring delays, we further add incidents on link travel times on different days. Specifically, we assume the probability of incident occurrence is 0.2 for each day. If an incident occurs on a particular day, the link travel times of the involved link over the planning horizon will be increased by multiplying the actual travel time by a random integer in set {2, 3}.

Consider an OD pair with potential multiple alternative routes, as shown in Figure 9. Note that the Lagrangian dual problem has introduced a large number of variables, namely, 8096 multipliers. A global search method is also used to seek the tightest bound of the prime problem, in which for each seed of initial multipliers, the inner loop of the sub-gradient algorithm will be repeated for ten iterations (i.e., $K_{\text{max}} = 10$). After executing the algorithm with 20 outer-loop iterations with different initial seeds of LR multipliers, the obtained best upper bound and tightest lower bound of the primal problem, respectively, are $Z^{*}_{\text{LET}} = 135.35$, $Z^{*}_{\text{LR}} = 126.55$. In comparison, the WAS method generates a lower bound of $Z^{*}_{\text{WAS}} = 125.39$ with a relative gap of 7.36%. The gap and relative gap are listed in Table 10.

### Table 10 Optimal Solution and Lower Bounds

<table>
<thead>
<tr>
<th>Value</th>
<th>$Z^{*}_{\text{LET}}$</th>
<th>$Z^{*}_{\text{LR}}$</th>
<th>$Z^{*}_{\text{WAS}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gap</td>
<td>135.35</td>
<td>126.55</td>
<td>125.39</td>
</tr>
<tr>
<td>Relative Gap</td>
<td>--</td>
<td>8.80</td>
<td>9.96</td>
</tr>
</tbody>
</table>

#### 6. Conclusions and Future Research

To solve critical path-finding problems in a time-dependent, stochastic transportation network, this paper formulated two equivalent optimization models with a scenario-based representation for capturing correlations of spatial and temporal link travel time. Aiming to obtain the a priori LET shortest paths, first, we investigated the unique path constraint across different solutions corresponding to random scenarios. Then, we proposed several equivalent reformulations to establish linear inequalities that can be easily dualized by a Lagrangian relaxation-based approach.

To search for high-quality solutions, the formulation was further decomposed into two simple sub-problems. Then, a sub-gradient algorithm was designed to seek the optimal solution of the corresponding Lagrangian dual problem. The effectiveness of the proposed approach was demonstrated by using a simplified network and two real-world networks. The experimental results on a medium-scale network
network extracted from the San Diego region show that the proposed algorithm can produce a smaller duality gap (about 2.57%) than the WAS. Even for the large-scale network in the Netherlands, the proposed Lagrangian relaxation-based approach generated reasonably tight lower bounds.

Further research will focus on the following three major aspects. (1) The proposed a priori optimal path modeling methodology will be further extended for a two-stage or multi-stage optimization model for emerging real-time adaptive routing applications. (2) The proposed model for the single OD shortest path problem in a time-dependent, stochastic transportation network can be further extended to a network-wide traffic assignment problem involving multiple OD pairs with stochastic demand patterns or road capacities. (3) To speed up the computational performance for large-scale real-world networks, improved reformulations or solution methods are also critically needed to aggregate or reduce the large number of Lagrangian relaxation multipliers for the proposed reformulation models.

Acknowledgments
The research of the first author was supported by the National Natural Science Foundation of China (No. 70901006) and the Research Foundation of State Key Laboratory of Rail Traffic Control and Safety, Beijing Jiaotong University (Nos. RCS2009ZT001, RCS2010ZZ001). We also want to thank Yusen Chen for providing the third network and sensor data. The work presented in this paper remains the sole responsibility of the authors.

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


