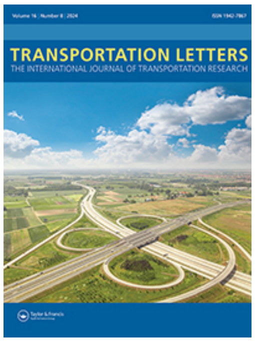

Taylor & Francis
Taylor & Francis Group

Transportation Letters

The International Journal of Transportation Research

ISSN: (Print) (Online) Journal homepage: www.tandfonline.com/journals/ytrl20

Quantum computing to solve scenario-based stochastic time-dependent shortest path routing

Vinayak V. Dixit, Chence Niu, David Rey, S. Travis Waller & Michael W. Levin

To cite this article: Vinayak V. Dixit, Chence Niu, David Rey, S. Travis Waller & Michael W. Levin (2024) Quantum computing to solve scenario-based stochastic time-dependent shortest path routing, Transportation Letters, 16:8, 793-803, DOI: 10.1080/19427867.2023.2238461

To link to this article: https://doi.org/10.1080/19427867.2023.2238461

Published online: 29 Jul 2023.

TRANSPORTATION LETTERS
2024, VOL. 16, NO. 8, 793–803
https://doi.org/10.1080/19427867.2023.2238461

Taylor & Francis
Taylor & Francis Group

OPEN ACCESS Check for updates

Quantum computing to solve scenario-based stochastic time-dependent shortest path routing

Vinayak V. Dixit[a], Chence Niu[a], David Rey[b], S. Travis Waller[a*] and Michael W. Levin[c]

[a]School of Civil and Environmental Engineering, UNSW Sydney, Sydney, Australia; [b]SKEMA Business School, Université Côte d'Azur, Sophia Antipolis, France; [c]School of Civil and Environmental Engineering, University of Minnesota, Minneapolis, USA

ABSTRACT
Networks are inherently uncertain and require scenario-based approaches to handle variability. In stochastic and time-dependent networks, optimal solutions cannot always be found using deterministic algorithms. Furthermore, Stochastic Time Dependent Shortest Path problems are known to be NP-hard. Emerging Quantum Computing Methods are providing new ways to address these problems. In this paper, the STDSP problem is formulated as a Quadratic Constrained Binary Optimization Problem. We show that in the case of independent link costs, the size of the problem increases exponentially. Finally, we find that using the quantum solver provides a linear computational experience with respect to the size of the problem. The proposed solution has implications for stochastic networks across different contexts including communications, traffic, industrial operations, electricity, water, broader supply chains, and epidemiology.

ARTICLE HISTORY
Received 10 April 2023
Accepted 15 July 2023

KEYWORDS
Stochastic-time-dependent shortest path; quantum annealing; quantum computing; constrained quadratic optimization model

Introduction

Routing in networks remains an active research area due to its multiple applications in telecommunications, transportation, industrial operations, electricity, water, broader supply chains, and epidemiology. Particularly in the area of logistics and supply chain where time-dependent routing (Yang et al. 2020) as well as stochastic optimization (for example (Huang, Huang, and Jian 2022)) play a significant role in improving efficiencies. Further, risks in networks are particularly prevalent owing to uncertainties ranging from demand and supply fluctuations. In this sense, it is reasonable to represent link travel time (costs) as a random variable that can take different values with a known probability distribution.

A priori routing with stochastically independent link costs has received a large share of attention: in this case, each link travel time random variable is independent of other link travel time random variables. Though this mathematical representation of uncertainty might seem general, however realistic networks demonstrate an underlying structure in the stochasticity. Networks with stochastically independent link costs can result in large number of potential network states with the realistic network states forming a significantly smaller subset.

We focus on an a priori, stochastic, and time-dependent routing problem where the representation of the uncertainty in link costs is based on mutually exclusive network states. We investigate the case where link costs are a function of the network state; which results in assuming that link costs are stochastically dependent. For instance, in the case of transportation networks if we consider two network states, e.g. sunny and rainy weather, it is reasonable to assume that the set of link travel times are different altogether. In other words, the travel time for each link depends on the arrival time at the start of that link. Additionally, for each time interval, we assume that the travel time for a link is random, meaning it can be represented by a random variable with a given probability distribution. It is important to note that there exist two mutually exclusive network states each with link travel time that are only a function of the conditions, i.e. sunny or rainy. This is referred to as Scenario-based Stochastic Time Dependent Shortest Path, or Scenario-based Least Expected Time (SLET) Path. This representation of uncertainty is fundamentally different from link costs that are stochastically independent of each other.

In this paper, we propose a Quadratic Constrained Binary Optimization Problem for the Scenario-based stochastic and time-dependent routing problem. We also show that in the case with independent random link costs, the size of the problem increases exponentially. By using quantum computers, more specifically quantum annealing, these problems are solved by the hybrid constrained quadratic model (CQM) solver provided by D-Wave. We start with introducing quantum computing as well as the stochastic time dependent shortest path problem.

Literature review

In this section, we summarize the current state of quantum computing particularly focusing on quantum annealing. We also summarize the main contributions on Shortest Path (SP) routing in networks with a focus on transportation-oriented applications and the representation of the uncertainty on the link costs therein.

Quantum computing

Research in quantum computing and algorithms over the past three decades have theoretically demonstrated the potential gains through 'quantum speedup' (Montanaro 2016). Quantum computing will have a disruptive impact on logistics and supply chain both from a computing and security perspective. At a fundamental level, quantum computers differ from classical computers in their ability

CONTACT Chence Niu chence.niu@unsw.edu.au School of Civil and Environmental Engineering, UNSW Sydney
*S. Travis Waller was affiliated with the School of Civil and Environmental Engineering, UNSW Sydney during the initial phase of this research, and he is now affiliated with the Faculty of Transport and Traffic Sciences, Lighthouse Professorship "Transport Modelling and Simulation", Technische Universität Dresden, Dresden, Germany.

to leverage quantum mechanical properties such as superposition, entanglement, and interference to speedup computations.

There has been ground breaking theoretical work that demonstrated quantum algorithms relying on quantum logic gates can provide significant speedups, one of the most celebrated being the Shor's algorithm (Shor 1994.), that demonstrated that quantum computers can solve the prime factorization problem exponentially faster than classical computers, having significant implications on cryptography. Recently 'Quantum Supremacy' was demonstrated on a problem that would take a classical supercomputer 10,000 years to be completed by 53 qubit Sycamore processor in 200 s (Arute et al. 2019). Applications of quantum algorithms in the field of transportation and traffic have been limited. Dixit and Jian (Dixit and Jian 2022a) used quantum gates for drive cycle analysis, that has applications to safety and emissions. Some other transportation problems such as traveling salesman problem, vehicle routing problem, traffic signals control problem, and traffic flow optimization problem have been investigated (Papalitsas et al. 2019; Warren 2020)– (Neukart et al. 2017).

Quantum computational engines based on quantum annealing are fundamentally different, for example D-Wave quantum computers. In classical computing, a bit can only be in one of two states – either 0 or 1. Therefore, two classical bits can represent four possible combinations: (0,0), (0,1), (1,0), and (1,1). However, in quantum computing, a qubit can be in a superposition of both 0 and 1 at the same time. This means that two qubits can represent four possible combinations of both 0 and 1, namely (0,0), (1,0), (0,1), and (1,1). The number of possible energy states grows exponentially with the number of qubits, which gives quantum computing significant advantages over classical computing. For example, a quantum computer with 20 qubits can represent 2^20 (or about 1 million) different energy states simultaneously. Therefore, quantum computing has the potential to perform certain types of calculations much faster than classical computing, because it can perform many calculations at the same time due to the exponential growth of energy states with the number of qubits. Quantum annealing uses quantum physics to find the solutions with the lowest energy states (the objective function). In the D-Wave quantum computer, the energy of each state is dependent on the biases of the qubits and the coupling between them which are decided by the problem formulation. The quantum annealing is to start from a particular system state to that of the final state defined by a Hamiltonian defining the feasible states. As is well known, finding minimum energy states in non-convex Hamiltonians is an NP-hard problem that classical computers take a long time to solve. A D-Wave's quantum annealer (QA) implements the optimization problem as a following time-dependent Ising Hamiltonian:

$$H_{Q_A}/h = \underbrace{-A\left(t/t_f\right)\sum_i \sigma_i^X}_{\textit{Initial Hamiltonian}} + \underbrace{B\left(t/t_f\right)\left(\sum_i h_i\sigma_i^Z + \sum_{i>j} J_{ij}\sigma_i^Z\sigma_j^Z\right)}_{\textit{Final Hamiltonian}} \tag{1}$$

where, t_f is the annealing time, σ_i^X and σ_i^Z are the Pauli matrices acting on qubit i, and h_i and J_{ij} are the qubit biases and coupling strengths, respectively. The quantum annealing begins with the ground state of the initial Hamiltonian, where all qubits are in a superposition of $|0\rangle$ and $|1\rangle$. Following a preset annealing schedule given by the time dependent functions $A\left(t/t_f\right)$ and $B\left(t/t_f\right)$, the Hamiltonian of the system slowly changes from the initial to the final Hamiltonian state, which encodes the solution of the given optimization problem. The Quantum Annealer solves Ising minimization problems, which are isomorphic to a Quadratic Unconstrained Binary Optimization (QUBO) Problem that are NP-Hard problems of the form:

$$\text{Obj} := x^T Q x \tag{2}$$

where x is a vector of N binary variables and Q is a NxN matrix representing the coefficients of the quadratic terms. The diagonal terms of Q are mapped to h_i and the cross terms are mapped to J_{ij} in the final Hamiltonian.

Recently, there are more studies focusing on the transport optimization problems solved by quantum computing, such as drive cycles estimation (Dixit and Jian 2022b), transport network design problems (Dixit and Niu 2022), traveling salesman problems (Jain 2021; Warren 2020), pre-production vehicle configurations problem (Glos, Kundu, and Salehi 2022), logistics optimization problem (Sales and Araos 2023), Vehicle Routing Problems as well as its variants such as multi-depot capacitated vehicle routing problem (MDCVRP) and its dynamic version (Harikrishnakumar et al. 2020), Traffic signal control (Hussain et al. 2020) and) Redistributing and rerouting vehicles for optimal network utilization (Neukart et al. 2017). Both the gate-based quantum computing with 15 qubits and quantum annealing with thousands of qubits are used to solve optimization problems. Compared with the traditional methods, quantum computing shows the potential of vast computing power and advanced capacities for large-scale problems. The problems are usually formulated as Quantum Fourier transform, QUBO and mixed integer programming model with different solvers. With more qubits and better connection, quantum computing is able to build a more efficient and sustainable transportation ecosystem.

Stochastic time dependent shortest path

Dreyfus (Dreyfus 1969) was the first to observe that Dijkstra's algorithm (Dijkstra 1959) could be used to find the Least Time (LT) path with the same (polynomial time) complexity, given an optimal waiting policy at intermediate nodes: such waiting policy assumes that a traveler always waits for the departure time which provides the earliest arrival time at the next node. However, the relaxation of this assumption had a severe impact on the complexity of time-dependent SP algorithms as observed by Orda and Rom (Orda and Rom 1990). In particular, finding that the LT path in a time-dependent network is known to be a NP-hard problem if the network is not First-In First-Out (FIFO) and if the waiting policy is not optimal (Sherali, Ozbay, and Subramanian 1998).

Finding the shortest paths (SP) in Stochastic Time Dependent (STD) networks was first studied by Hall (Hall 1986) who showed that the generic SP algorithm used in stochastic time invariant networks cannot always find the LET path in STD networks. Psaraftis and Tsitsiklis (Psaraftis and Tsitsiklis 1993) studied a variant of the stochastic and time-dependent SP problem where the state of successive arcs is revealed upon arrival at a node and consider a routing policy where waiting time at nodes is permitted but penalized. They were able to obtain efficient algorithms given the assumption that the network is acyclic. Miller-Hooks and Mahmassani (Miller-Hooks and Mahmassani 2000) provided an optimal procedure with FIFO and no-waiting policy to find the LET path, and introduced the notion of label dominance for STD networks. In order to ensure global optimality, the proposed algorithm must maintain an arbitrarily large but finite list of nondominated labels at each node of the network. As the number of nondominated labels at a node can grow exponentially (depending on the number of paths traveling through the node of interest), the aforementioned algorithm has a super polynomial time-complexity. Ezaki et al (Ezaki, Imura, and Nishinari 2022)., designed an adaptive route-finding algorithm to find the fastest path for square lattice, random, and hub-and-spoke networks under different demand scenarios. Bertsimas et al (Bertsimas, Jaillet, and Odoni 1990)., proposed a priori optimization strategies that utilizing an updating method to minimize the expected cost of the priori solution. A review of the online routing variants in stochastic, time-dependent networks is provided by Gao and Chabini (Gao and Chabini 2006) who propose a two-dimensional taxonomy of STD routing problems according to link cost dependency and information access. Their findings confirm that the assumptions in the modeling of stochastic, time-dependent networks can have a significant impact on the computational efficiency of routing algorithms.

The concept of scenario-based representation has emerged as a promising method to systematically incorporate these uncertainties into transport models (Hamdouch et al. 2004; Marcotte et al. 2004; Dixit, Gardner, and Waller 2013)– (Waller et al. 2013). In a scenario-based representation of the correlations among link travel times, each combination of time-dependent link costs corresponds to a state of the network, e.g. adverse weather, time of the week, etc.; which can be realized with a known probability. Recognizing inherent correlations in link travel times, Nie and Wu (Nie and Wu 2009) attempted to account for correlations between adjacent arcs in STD networks to find the most reliable path using label correcting algorithms. Huang and Gao (Huang and Gao 2012) proposed an algorithm to find the LET paths in STD networks with stochastically dependent arcs that was consistent with the scenario-based stochastic representation. They establish and use path dominance conditions over the product set of time intervals and scenarios (support points) and show that, under these conditions, the extension of Bellman's principle of optimality as presented by (Miller-Hooks and Mahmassani 2000) does not hold and thus, for every candidate optimal path, and sub-path non-dominance properties must be checked, making the computational experience exponential.

Currently available methods often require long computation times, and as the size of the problem increases, the computation time grows exponentially. In this paper, we focus on the case where the network is time-dependent and the stochastic network states are mutually exclusive, hereby referred to as scenario-dependent arcs. We formulate the problem as a Quadratic Constrained Binary Optimization Problem., which makes a quantum annealing-based quantum computing method an appealing solution method. We also formulate the case for independent stochastic links and show that the size of the variables increases exponentially. We show that the computational experience with quantum annealing is linear as compared to other classical approaches that are exponential (Huang and Gao 2012).

Mathematical model

We seek to find the shortest path between two points (vertices) in a transport network where the travel time on each link (arc) of the route is assumed to be time-dependent, i.e. the travel time on each link depends on the arrival time at the beginning of the link. Further, for each time interval, we assume that the travel time on a link is stochastic, i.e. the travel time can be represented by a random variable with a given probability distribution. This model for travel time variability aims to represent a congestion episode in a transport network during which we seek to find the expected fastest route between two locations. We use graph theory to represent mathematically this routing problem and the notation used throughout the paper is summarized in Table 1. The set V represents nodes in the network and the set A represents the physical links between these nodes. We assume that time can be discretized into elementary time intervals and T is set of all the available time intervals. The set K is the set of network states used to account for different realization of the travel demand. Using this notation, the graph $G = (V, A, T, K, [\tau_{ij}^k(t)])$ is a STD network where link weights represent time-dependent travel times with a known probability distribution.

Table 1. Mathematical notation.

V	Set of vertices
A	Set of arcs
T	Set of time intervals
K	Set of scenarios
p_k	Probability that scenario k is realized
$\tau_{ij}^k(t)$	Travel time on $(i,j) \in A$ if one departs$i$ at time t in scenario k
$x_{ij}^k(t)$	Binary variable for link $(i,j) \in A$ in scenario k, at time t, if it is part of SLET
$\delta^+(i)$	is the set of all nodes j such that $(i,j) \in A$
$\delta^-(j)$	is the set of all nodes i such that $(i,j) \in A$
ξ^*	The minimal expected travel cost of the naive algorithm (Algorithm 1)

In contrast to previous research on this STD routing problem, we focus on the case where the travel demand scenarios are mutually exclusive. This is consistent with the scenario-based paradigm where each scenario is representative of a network state and no two states can be realized simultaneously. Namely, we assume that the network state can be represented by a discrete random variable over the set of scenarios K with a Probability Mass Function (PMF) p_k, $\forall k \in K$ and $\sum_{k \in K} p_k = 1$. Under the assumption that network states are mutually exclusive, it results that link travel time are stochastically dependent, i.e. only a single network state is realized and the probability that two or more link travel time from different scenarios are jointly realized is zero. Therefore, the joint probability that two different scenarios are realized is assumed null, i.e. $p_k p_{k'} \equiv 0$. We summarize this property of STD networks in the following definition.

Definition 1 (Scenario-based Network).Let G be a STD network. If no two scenarios can be jointly realized, we say that G is a scenario-based STD network.

The routing problem of interest in this paper can then be stated as follows: given a scenario-based STD network, find the Scenario-based Least Expected Time (SLET) path from an origin node to a destination node. Throughout this paper we assume that the source node s and destination node d are fixed and thus seek the SLET path from s to d.

In the context of scenario-based STD networks, since stochastic scenarios are mutually exclusive each network state can be represented by a deterministic and time-dependent network and traditional node-labels can be used to find the LT path within each scenario. In order to clarify the difference between the different types of node-labels in our problem, we introduce the following definitions.

Let $\lambda_i^{\mu k}(t)$ be the travel time from node i at time t through path μ to the destination node in scenario k. Accordingly, for a set of stochastic scenarios K, the expected-node-cost $\kappa_i^{\mu}(t)$ from i to the destination node at time t through path μ can be determined recursively by the formula:

$$\kappa_i^\mu(t) = \mathrm{E}\left[\lambda_i^\mu(t)\right] = \sum_{k \in K} p_k \lambda_i^{\mu k}(t) \tag{3}$$

The optimality conditions for the SLET path for each node and time interval are:

$$\kappa_i(t) = \min_\mu\left\{\kappa_i^\mu(t)\right\} \ \forall i, \forall t \tag{4}$$

A STD network with independent link probabilities each with K states, can be trivially represented as a Scenario-based STD network with a convolution of K states. In a STD network with independent link probabilities with K states, a path μ will have states that are a convolution of which can be trivially represented as a Scenario-based STD network with a convolution of K state across all edges and time periods, i.e. it corresponds to a maximum of $|K|^{|A|+|T|}$mutually exclusive network states. This is demonstrated in Figure A1. A detailed example is discussed in the Appendix.

Current approaches to determine SLET paths rely on tracking node labels and evaluating it against a dominance criteria, the best-known approach relies on TS dominance by Huang and Gao (2012) which is known to be exponential. Our paper contributes to this literature by formulating the SLET problem as a mixed-integer programming model which is solved by the Hybrid CQM solver. We benchmark the performance of the quantum annealing approach with classical approaches.

The Quadratic Constrained Binary Optimization Problem for finding the SLET is given by Equations (5)-(10). The definition of the variables is provided in Table 1. The objective function defined in Equation (5) minimizes the expected path costs and determines when a link is part of a SLET in each scenario.

Equation (6) ensures that at time 0 at least one of the outbound links from the source is used. Equation (7) represents the constraint enforcing that one link to the destination should be used across all time periods, i.e. the destination node should be reached across all the time periods.

Equation (8) ensures that the same path is used across different scenarios. This is achieved by ensuring that if a link is used at anytime in one scenario, then it should be used in every other scenario, this is represented as the sum of the link variable across all time should be equal for all scenarios.

Equation (9) ensures that the sum of link incidences at a node at time t is equal to the sum of outbound link incidences. This enforces time dependency and that there is no waiting at a node. Since there is no quadratic term in the above equations, the proposed model can be considered as a mixed integer linear programming (MILP) model.

$$\min \sum_k p^k \sum_t \sum_{(ij)\in A} \tau_{ij}^k(t) x_{ij}^k(t) \tag{5}$$

$$\sum_{i \in \delta^+(s)} x_{si}^k(0) = 1 \forall k \tag{6}$$

$$\sum_t \sum_{i \in \delta^-(d)} x_{id}^k(t) = 1 \forall k \tag{7}$$

$$\sum_t x_{ij}^l(t) = \sum_t x_{ij}^m(t) \forall l, m \in \mathbf{K}; \forall (ij) \in A \tag{8}$$

$$\sum_{i \in \delta^-(j)} x_{ij}^k(t')\Big|_{t' + \tau_{ij}^k(t') = t} - \sum_{l \in \delta^+(j)} x_{jl}^k(t) = 0 \ \forall k, t, \forall j \in V/\{d\} \tag{9}$$

$$x_{ij}^k(t) \text{ is a binary variable } \forall \text{t}; \forall \text{k}; \forall (ij) \in A \tag{10}$$

Quantum annealing

We implement the STDSP on a D-Wave Advantage quantum computer. The D-Wave's Quantum Processing Unit (QPU) with only 5,436 qubits is connected based on the Pegasus topology, however, several qubits are not connected. The limited number of qubits and connectivity creates significant challenges to embed and solve large problems. The D-Wave system attempts to navigate the connectivity issue by copying an optimization variable to multiple qubits, which is also referred to as *chain strength*, which not only reduces the number of qubits, but also errors in one qubit propagates significantly to affect the quality of the solution. This embedding is done automatically by the D-Wave system.

We solve the network in Example 1, which has 42 variables, on the D-Wave Advantage Quantum Computer with 150 samples required 0.0344 s of QPU time. For benchmarking, the SLET was calculated solving the QUBO using CPLEX on a Win64 i7 machine with 2.9 GHz and 8Gb of RAM, which took approximately 0.02 s.

It is important to note that a CPU is able to use multiple cores to solve a problem by executing a series of instructions, each taking about a nanosecond, while a QPU solves the same problem as a single instruction in microseconds. We do however expect that as the size of QPUs increases as well as the connectivity and its control improve, we could solve large problems quickly on these machines.

Given the current limitations in size of QPUs to solve large problems, D-Wave has implemented hybrid solvers that uses a combination of classical solvers and QPU. As shown in Figure 1, the problem is computed by a number of heuristic solvers to search for good-quality solutions by state-of-the-art CPU and/or GPU parallelly. Then the quantum modules (QM) formulate and send quantum queries to a D-Wave QPU which guides the heuristic search and improve the solution quality (Hall 1986). For comparing

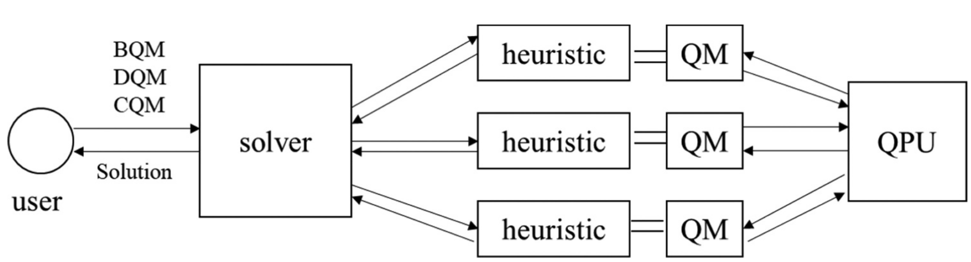

Figure 1. Structure of a hybrid solver in hybrid solver service (D-wave 2021).

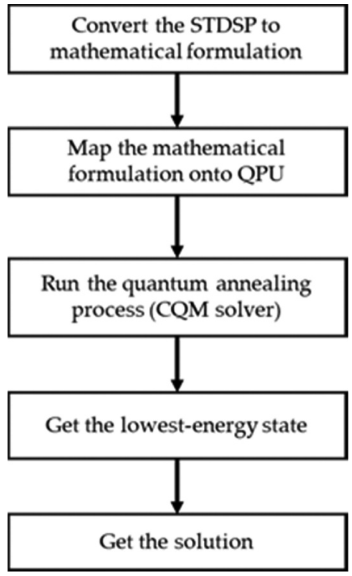

Figure 2. The flowchart of the solution procedure.

the computational experience, we use Hybrid Constrained Quadratic Model Version 1,[1] for binary problems. The flowchart of the solution procedure is shown in Figure 2. It should be noted that there may be multiple solutions to the problem for the CQM solver (D-wave 2021).

As discussed earlier, the best-known algorithm to solve for SLET by Huang and Gao (Huang and Gao 2012) is exponential. We explore the computational experience and trends by using D-Wave's hybrid solver with using CPLEX on a Win64 i7 machine with 3.40 GHz (8 threads) and 16Gb of RAM.

Results

In this section, we carry out the implementation of the SLET algorithm on benchmark networks to compare computational experiences.

Benchmark networks

In this benchmark, we implement the SLET algorithm on acyclic $n \times n$ grid networks, as depicted by Figure 3. For each type of network, we seek the SLET path from the source node in the top left corner to the destination node in the bottom-right node. We choose grid sizes: $n = 8$ (64 nodes, Figure 3) and 10 (100 nodes). For each link of the network and each time interval, 10, 15, and 20 scenarios and uniform scaled link travel times are generated. Three instances of each grid are generated, hence a total of 18 benchmark networks.

Besides the CPLEX results, a simple naive algorithm is proposed to measure the QPU performance. As shown in Algorithm1, there are different shortest paths under different scenarios and the minimal expected travel cost in the shortest paths pool is calculated to benchmark the QPU performance.

Algorithm 1: Naive algorithm

Input: $V, A, T, K, p_k, \tau_{ij}^k(t)$
Output: ξ^*
$\xi \leftarrow \emptyset$

(Continued)

Algorithm 1: Naive algorithm

Input: $V, A, T, K, p_k, \tau_{ij}^k(t)$
Output: ξ^*
$\xi \leftarrow \emptyset$
for $k = 1..|K|$ **do**
  $X^k \leftarrow$ the shortest path under the scenario k
  $E(X^k) = \sum_k p^k \sum_t \sum_{(ij) \in A} \tau_{ij}^k(t) x_{ij}^k(t) \text{where} x_{ij}^k(t) \in X^k \leftarrow$ the expected path cost of X^k under all the scenarios
  $\xi = \xi \cup^E (X^k)$
$\xi^* = \min \xi$

$$\text{Time CPU} = 3.086 e^{5E-05 \times |A| \times K \times T}, R^2 = 0.982$$

$$\text{Time Leap Hybrid} = 0.0014 \times |A| \times K \times T, \; R^2 = 0.649$$

$$\text{a} : \Delta \text{ Obj.}\% = \left(\text{Travel cost}_{\text{Hybrid}} - \text{Travel cost}_{\text{CPLEX}}\right) / \text{Travel cost}_{\text{CPLEX}}$$

$$\text{b} : \Delta \text{ Time. }\% = \left(\text{Time}_{\text{Hybrid}} - \text{Time}_{\text{CPLEX}}\right) / \text{Time}_{\text{CPLEX}}$$

$$\text{c} : \Delta \text{ Obj.}\% = \left(\text{Travel cost}_{\text{Hybrid}} - \text{Travel cost}_{\text{Naive}}\right) / \text{Travel cost}_{\text{Naive}}$$

As discussed earlier, quantum computers relies on providing sufficient time for the efficient solutions to emerge. Based on iteratively setting a threshold sampling time such that the efficient solutions were found for all the cases. This threshold sampling time was found to be $0.0014 \times |A| \times K \times T$. Therefore, the computational experience was found to be linear for the domain of problems tested in this paper.

As shown in Table 2, for the domain of problems tested, the CPLEX solvers is faster in terms of absolute run times when the number of variables is small (roughly less than 60,000). However, the CPU run times with a CPLEX solver increases exponentially as shown in Figure 4. This is expected given the NP-hardness of

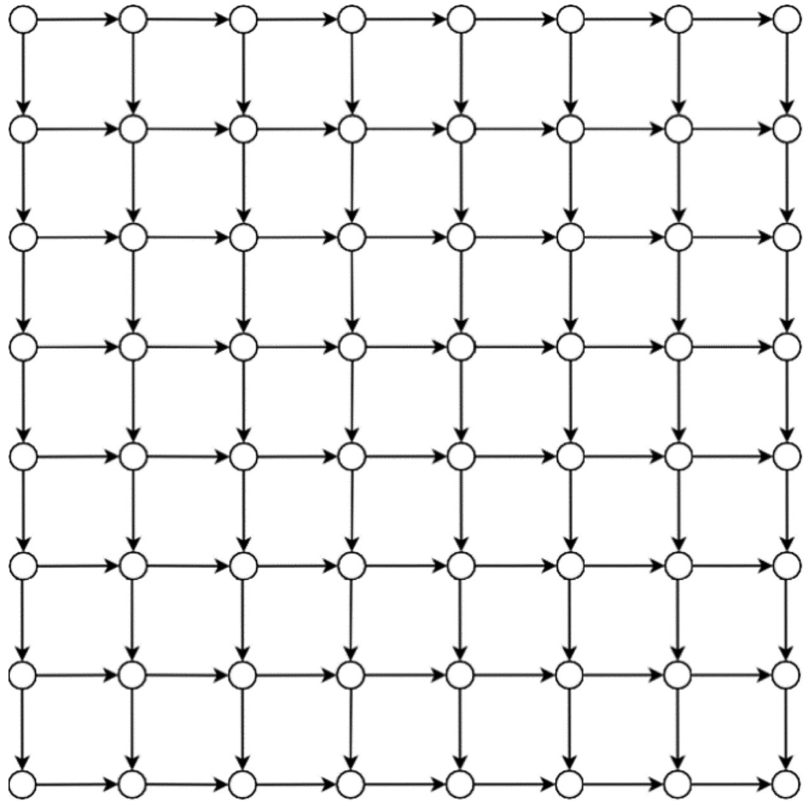

Figure 3. An example of benchmark grid networks (8x8 acyclic network).

Table 2. Computational results of benchmark networks.

				CPLEX		Hybrid-CQM Solver				Naive	
Grid	Scenario	Time	Variables	Min Obj.	Time (s)	Min Obj.	Time (s)	ΔObj.%[a]	ΔTime.%[b]	Min obj.	ΔObj.%[c]
				19.40	10.14	19.79	37.76	2.01%	272.28%	20.00	−1.05%
8	10	30	34720	19.30	9.09	19.79	24.86	2.54%	173.38%	20.00	−1.05%
				19.10	11.96	19.19	74.71	0.47%	524.62%	19.90	−3.57%
			52080	19.40	34.27	19.79	59.18	2.01%	72.69%	20.00	−1.05%
8	15	30		19.60	46.15	19.73	44.97	0.68%	−2.55%	20.07	−1.69%
				19.67	47.42	20.00	55.42	1.69%	16.87%	20.13	−0.65%
			69440	19.80	65.42	19.95	134.06	0.76%	104.91%	19.85	0.50%
8	20	30		19.95	98.28	20.30	57.53	1.75%	−41.47%	20.15	0.74%
				19.45	107.93	20.00	58.65	2.83%	−45.65%	19.85	0.76%
				24.60	206.16	25.40	52.09	3.25%	−74.73%	25.50	−0.39%
10	10	40	73800	25.00	151.75	25.89	91.39	3.56%	−39.77%	26.20	−1.18%
				24.30	62.89	25.20	48.69	3.70%	−22.58%	25.20	0.00%
				25.07	542.23	25.93	119.48	3.44%	−77.97%	25.87	0.23%
10	15	40	110700	24.73	640.00	25.59	100.79	3.46%	−84.25%	25.53	0.24%
				25.47	563.41	26	214.7	2.08%	−61.89%	26.00	0.00%
				25.40	2678.62	25.80	287.00	1.57%	−89.29%	25.85	−0.19%
10	20	40	147600	24.95	2169.00	26.10	145.03	4.61%	−93.31%	26.20	−0.38%
				25.45	2791.26	26.20	161.93	2.95%	−94.20%	26.15	0.19%

integer programming, and is similar to the computational experience reported in (Huang and Gao 2012) when using their label correcting algorithm. However, the computation times required by quantum computing appears to follow a linear trend and is orders of magnitude faster than CPLEX solver, with a maximum is 17 times faster. In terms of the solution quality, the CPLEX solver is better than CQM solver, but the relative gap is small, ranging from 0.47% to 4.61%. Compared to the naive algorithm, the CQM solver has a higher quality of solution in most cases. In 6 of 18 cases the solution quality of CQM solver is worse than the naive algorithm, with an absolute difference between 0.05 to 0.15. Quantum computing has demonstrated advantages in large-scale computation, with significant reductions in computation time compared to CPLEX solver. While the quality of some solutions may not be as good as traditional methods, the solution quality is still acceptable.

Conclusions

In this paper, we develop a mixed-integer programming model to find the SLET paths in STD networks where the network states are assumed to be mutually exclusive, i.e. the link travel times are assumed to be scenario-dependent. We also demonstrate that an

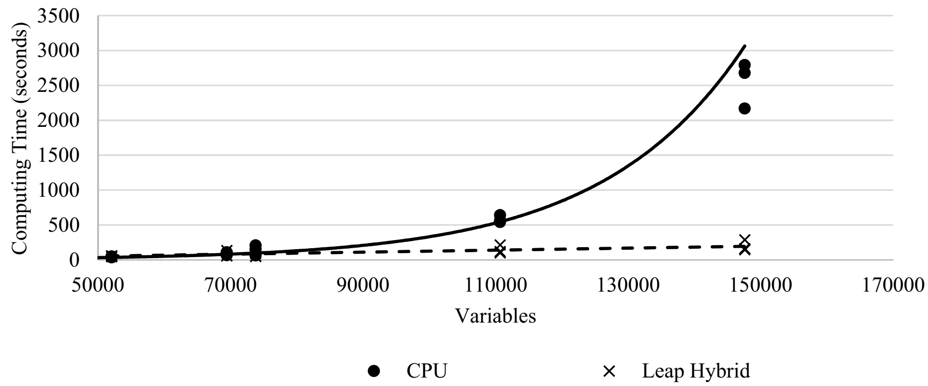

Figure 4. Computational experience.

STD network with independent arc links can also be represented as mutually exclusive scenarios that increase exponential with the number of stochastic links. Our approach is motivated by the observation that in most networks the representation of the uncertainty on links can be organized based on exogenous scenarios, which would be a subset of the scenarios generated with the independent links.

We introduce an elementary example that demonstrates the behavior of the proposed approach and illustrates differences between STD networks with stochastic scenarios and stochastically independent link travel times. We then solve this example on a D-Wave quantum processing unit.

To our best knowledge, this is the first work about quantum computing on STDSP problems. The emerging quantum computing has a potential advantage in the field of transportation optimization, where the system is characterized by an extensive network of interdependent variables and numerous constraints. At the early stage of quantum computing, it is worth exploring and comparing the performance between classical and quantum computing. Quantum computing displays a noteworthy capacity to deliver efficient solutions within a manageable time frame. This proficiency is particularly pronounced in handling large-scale problems. Unlike traditional CPLEX solvers, which follow an exponential computational trend, quantum computing exhibits a linear trajectory in computational duration, a clear testament to its efficiency in complex computations. While CPLEX solvers tend to yield faster results for small-scale problems (less than 60,000 variables), quantum computing outshines its traditional counterpart in larger problem sets. Specifically, it is up to 17 times more efficient than CPLEX solvers, underlining the power of quantum computing in managing intricate computations. While the quality of solutions from the CPLEX solver is slightly superior to those from the quantum method (CQM solver), the gap is relatively minor, ranging from 0.47% to 4.61%. Despite certain solution quality shortcomings, the overall performance of quantum computing stays within an acceptable range, demonstrating its viability as a computational tool.

Due to the limitation of the number of qubits, we only apply the hybrid solver for quantum computing to solve the STDSP problem. As quantum computing is still in the Noisy Intermediate Scale Quantum (NISO) (Preskill 2018; Torlai and Melko 2020) era, quantum systems are highly susceptible to environmental noise and errors. The challenge lies in striking the optimal balance between solution quality and computational efficiency, an equilibrium crucial in solidifying quantum computing as a reliable tool for large-scale computational problem-solving. Therefore, ensuring accurate results necessitates the development of robust error mitigation strategies. It is worth noting that with more qubits, better connectivity and error correction we can see faster and more reliable QPU in the near future, that can solve these problems orders of magnitude faster. It is able to explore the potential of emerging quantum computing for large-scale traffic problems by comparing pure quantum computing with traditional methods.

Note

1. The hybrid CQM solver is able to tune the parameters such as number of reads and penalty weights automatically. The inner specific methods are not known.

Acknowledgments

This research was supported under the IAG Chair Position. I would also like to thank Dr. Victoria Goliber and Mr. Irwan Owen from DWAVE for their continued support from a technical and logistics standpoint.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Author contributions

Conceptualization, V.D., D.R. and T.W.; methodology, V.D.; software, V.D., C. N., and M.L.; validation, V.D., M.L. and C.N.; resources, V.D.; data curation, C. N.; writing – original draft preparation, V.D.; writing – review and editing, V.D., C.N., D.R., M.L. and T.W.

Data availability statement

The data are available from the corresponding author upon reasonable request.

References

Arute, F., K. Arya, R. Babbush, D. Bacon, J. C. Bardin, R. Barends, R. Biswas, et al. 2019. "Quantum Supremacy Using a Programmable Superconducting Processor." *Nature* 574 (7779): 505–510. https://doi.org/10.1038/s41586-019-1666-5.

Bertsimas, D. J., P. Jaillet, and A. R. Odoni. Dec 1990. "A Priori Optimization." *Operations Research* 38 (6): 1019–1033. https://doi.org/10.1287/opre.38.6.1019.

Dijkstra, E. W. 1959. "A Note on Two Problems in Connexion with Graphs." *Numerische Mathematik* 1 (1): 269–271)," 1959. (Numerische Mathematik. https://doi.org/10.1007/BF01386390.

Dixit, V., L. M. Gardner, and S. T. Waller. 2013. "Strategic User Equilibrium Assignment Under Trip Variability." In *Transportation Research Board 92nd Annual Meeting*, Washington, DC, USA, vol. 9.

Dixit, V., and S. Jian. 2022a. "Quantum Fourier Transform to Estimate Drive Cycles." *Scientific Reports* 12 (1): 1–10. https://doi.org/10.1038/s41598-021-04639-0/.

Dixit, V., and S. Jian. Jan 2022b. "Quantum Fourier Transform to Estimate Drive Cycles." *Scientific Reports* 12 (1): 654. https://doi.org/10.1038/s41598-021-04639-0 .

Dixit, V., and C. Niu. 2022. "Quantum Computing for Transport Network Design Problems." *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.4016306.

Dreyfus, S. E. 1969. "An Appraisal of Some Shortest-Path Algorithms." *Operations Research* 17 (3): 395–412. https://doi.org/10.1287/opre.17.3.395.

D-wave, "Hybrid Solver for Constrained Quadratic Models," 14-1055A–A, Oct. 2021. [Online]. Available: https://www.dwavesys.com/media/rldh2ghw/14-1055a-a_hybrid_solver_for_constrained_quadratic_models.pdf.

Ezaki, T., N. Imura, and K. Nishinari. 2022. "Towards Understanding Network Topology and Robustness of Logistics Systems." *Communications in Transportation Research* 2:100064. https://doi.org/10.1016/j.commtr.2022.100064. Dec.

Gao, S., and I. Chabini. Feb 2006. "Optimal Routing Policy Problems in Stochastic Time-Dependent Networks." *Transportation Research Part B: Methodological* 40 (2): 93–122. https://doi.org/10.1016/j.trb.2005.02.001 .

Glos, A., A. Kundu, and Ö. Salehi, "Optimizing the Production of Test Vehicles Using Hybrid Constrained Quantum Annealing," *arXiv: 2203.15421 [Quant-Ph]*, Mar. 2022, Accessed June 19, 2023. Available: http://arxiv.org/abs/2203.15421.

Hall, R. W. 1986. "The Fastest Path Through a Network with Random Time-Dependent Travel Times." *Transportation Science* 20 (3): 182–188. https://doi.org/10.1287/trsc.20.3.182.

Hamdouch, Y., P. Marcotte, and S. Nguyen. 2004. "A Strategic Model for Dynamic Traffic Assignment." *Networks and Spatial Economics* 4 (3): 291–315. https://doi.org/10.1023/B:NETS.0000039784.00352.66.

Harikrishnakumar, R., S. Nannapaneni, N. H. Nguyen, J. E. Steck, and E. C. Behrman. 2020. "A Quantum Annealing Approach for Dynamic Multi-Depot Capacitated Vehicle Routing Problem." *arXiv Preprint arXiv: 2005 12478*. https://doi.org/10.48550/arXiv.2005.12478.

Huang, H., and S. Gao. Jun 2012. "Optimal Paths in Dynamic Networks with Dependent Random Link Travel Times." *Transportation Research Part B: Methodological* 46 (5): 579–598. https://doi.org/10.1016/j.trb.2012.01.005.

Huang, W., W. Huang, and S. Jian. 2022. "One-Way Carsharing Service Design Under Demand Uncertainty: A Service Reliability-Based Two-Stage Stochastic Program Approach." *Transportation Research Part E: Logistics & Transportation Review* 159:102624. https://doi.org/10.1016/j.tre.2022.102624.

Hussain, H., M. B. Javaid, F. S. Khan, A. Dalal, and A. Khalique. 2020. "Optimal Control of Traffic Signals Using Quantum Annealing." *Quantum Information Processing* 19 (9): 1–18. https://doi.org/10.1007/s11128-020-02815-1.

Jain, S. 2021. "Solving the Traveling Salesman Problem on the D-Wave Quantum Computer." *Frontiers of Physics* 9:760783. https://doi.org/10.3389/fphy.2021.760783. Nov.

Marcotte, P., S. Nguyen, and A. Schoeb. 2004. "A Strategic Flow Model of Traffic Assignment in Static Capacitated Networks." *Operations Research* 52 (2): 191–212. https://doi.org/10.1287/opre.1030.0091.

Miller-Hooks, E. D., and H. S. Mahmassani. 2000. "Least Expected Time Paths in Stochastic, Time-Varying Transportation Networks." *Transportation Science* 34 (2): 198–215. https://doi.org/10.1287/trsc.34.2.198.12304 .

Montanaro, A. 2016. "Quantum Algorithms: An Overview." *Npj Quantum Information* 2 (1): 1–8. https://doi.org/10.1038/npjqi.2015.23.

Neukart, F., G. Compostella, C. Seidel, D. Von Dollen, S. Yarkoni, and B. Parney. 2017. "Traffic Flow Optimization Using a Quantum Annealer." *Frontiers in ICT* 4:29. https://doi.org/10.3389/fict.2017.00029.

Nie, Y. M., and X. Wu. 2009. "Reliable a Priori Shortest Path Problem with Limited Spatial and Temporal Dependencies." In *Transportation and Traffic Theory 2009: Golden Jubilee*, 169–195. Springer US. https://doi.org/10.1007/978-1-4419-0820-9_9.

Orda, A., and R. Rom. 1990. "Shortest-Path and Minimum-Delay Algorithms in Networks with Time-Dependent Edge-Length." *Journal of the ACM* 37 (3): 607–625. https://doi.org/10.1145/79147.214078.

Papalitsas, C., T. Andronikos, K. Giannakis, G. Theocharopoulou, and S. Fanarioti. 2019. "A QUBO Model for the Traveling Salesman Problem with Time Windows." *Algorithms* 12 (11): 224. https://doi.org/10.3390/a12110224.

Preskill, J. 2018. "Quantum Computing in the NISQ Era and Beyond." *Quantum* 2:79. https://doi.org/10.22331/q-2018-08-06-79. Aug.

Psaraftis, H. N., and J. N. Tsitsiklis. 1993. "Dynamic Shortest Paths in Acyclic Networks with Markovian Arc Costs." *Operations Research* 41 (1): 91–101. https://doi.org/10.1287/opre.41.1.91.

Sales, J. F. A. and R. A. P. Araos, "Adiabatic Quantum Computing for Logistic Transport Optimization," *arXiv: 2301.07691 [Quant-Ph]*, Jan. 2023, Accessed May 30, 2023. Available: http://arxiv.org/abs/2301.07691.

Sherali, H. D., K. Ozbay, and S. Subramanian. 1998. "The Time-Dependent Shortest Pair of Disjoint Paths Problem: Complexity, Models, and Algorithms." *Networks: An International Journal* 31 (4): 259–272. https://doi.org/10.1002/(SICI)1097-0037(199807)31:4<259:AID-NET6>3.0.CO;2-C.

Shor, P. W. 1994."Algorithms for Quantum Computation: Discrete Logarithms and Factoring." In *Proceedings 35th annual symposium on foundations of computer science*, Santa Fe, NM, USA, 124–134. IEEE.

Torlai, G., and R. G. Melko. 2020. "Machine-Learning Quantum States in the NISQ Era." *Annual Review of Condensed Matter Physics* 11 (1), Mar: 325–344. https://doi.org/10.1146/annurev-conmatphys-031119-050651 .

Waller, S. T., D. Fajardo, M. Duell, and V. Dixit. 2013. "Linear Programming Formulation for Strategic Dynamic Traffic Assignment." *Netw Spat Econ* 13 (4), Dec: 427–443. https://doi.org/10.1007/s11067-013-9187-5.

Warren, R. H. 2020. "Solving Traveling Salesman Problems by D-Wave Hybrid." 3. https://bpb-us-e1.wpmucdn.com/sites.psu.edu/dist/f/121688/files/2020/08/Solving-TSPs-by-D-Wave-Hybrid-1.pdf.

Yang, S., L. Ning, P. Shang, and L. C. Tong. 2020. "Augmented Lagrangian Relaxation Approach for Logistics Vehicle Routing Problem with Mixed Backhauls and Time Windows." *Transportation Research Part E: Logistics & Transportation Review* 135:101891. https://doi.org/10.1016/j.tre.2020.101891.

Appendix

Example 1. We consider a six-node and eight-link network with five paths (P1, P2, P3, P4, and P5) as depicted in Figure A1 Two stochastic *scenarios* $K = \{1, 2\}$ are used to represent the uncertainty on link costs, and the scenarios $K = \{1 \ldots 32\}$ represent the expanded mutually exclusive scenarios for the *independent stochastic* link costs. We seek to find the SLET path from the origin node S to the destination node D, for the two stochastic scenario ($K = \{1, 2\}$) cases. For the case with independent stochastic links the Least Expected Time (LET) is determined from s to d. The link travel times are given in Table A1. for specific time intervals, and we assume that the link travel times for each other time interval is equal to the nearest specified time interval.

Table A2 gives the path travel times and path expected travel times at time interval $t = 0$ and indicates the fastest path for the scenario-dependent arcs (SLET) across Scenarios 1 and 2, and the LET across the independent stochastic arcs. Assume that the PMF of the network state random variable is uniform, i.e. $p_1 = p_2 = 0.5$. The SLET path at $t = 0$ is P1 with an expected travel time of 2.5. Observe that the SLET path is not necessarily the LET path which is P3 with an expected travel time of 3. Example 1 highlights that the case with independent stochastic arc links can be represented as a set of mutually exclusive scenarios that has a cardinality of $|K|^{|A|+|T|}$ (i.e., $2^5 = 32$). Furthermore, it is important to note that the SLET path in a STD network can be different than the LET paths within independent stochastic arc links.

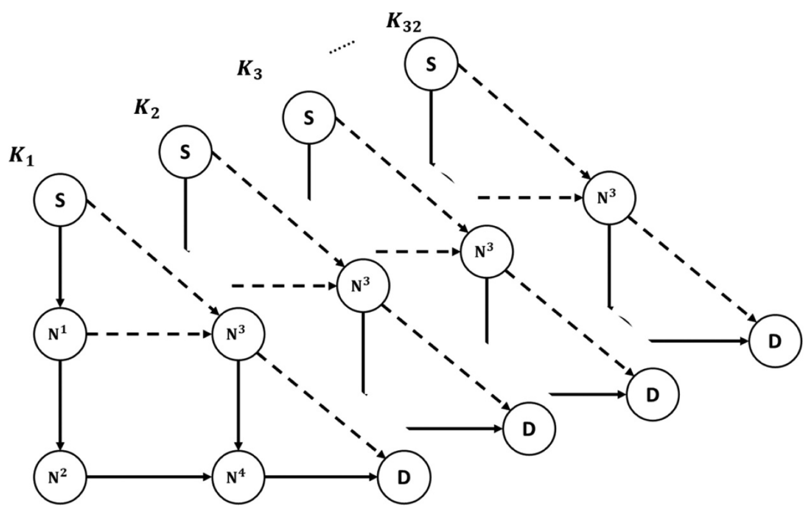

Figure A1. A six node and eight-link network with five paths.

Table A1. Link costs.

Time	Link	Scenarios
a: Links with deterministic costs		
Time	$S \rightarrow N^1$	1
	$N^1 \rightarrow N^2$	1
	$N^2 \rightarrow N^4$	1
	$N^3 \rightarrow N^4$	1
	$N^4 \rightarrow D$	1

		Independent scenarios																															
		Scenarios		Additional expanded scenarios representing the entire set of scenarios generated from Independent Links																													
Time	Link	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
b: Links with stochastic costs by Scenarios and Expansion of Independent Stochastic Links to Mutually Exclusive Scenarios																																	
0	$S \rightarrow N^3$	1	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
	$N^1 \rightarrow N^3$	1	100	1	1	1	1	1	1	1	100	100	100	100	100	100	100	100	1	1	1	1	1	1	1	1	100	100	100	100	100	100	100
	$N^3 \rightarrow D$	1	100	1	1	1	100	100	100	100	1	1	1	1	1	100	100	100	1	1	1	1	100	100	100	100	1	1	1	1	1	100	100
1	$S \rightarrow N^3$	1	2	1	2	2	1	1	2	2	1	1	2	2	2	1	2	2	1	1	2	2	1	1	2	2	1	1	2	2	2	1	2
	$N^1 \rightarrow N^3$	1	100	100	1	100	1	100	1	100	1	100	1	1	100	1	1	100	1	100	1	100	1	100	1	100	1	100	1	1	100	1	1
	$N^3 \rightarrow D$	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	$S \rightarrow N^3$	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	$N^1 \rightarrow N^3$	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	$N^3 \rightarrow D$	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Table A2. Path costs.

Departure time	Path ID	Path	Scenario-based (Scenarios 1–2)	Independent (Scenarios 1–32)
0	P1	$S \to N^3 \to$D	**2.5 (SLET)**	27.25
	P2	$S \to N^3 \to N^4 \to D$	3.5	3.5
	P3	$S \to N^1 \to N^3 \to D$	3	**3 (LET)**
	P4	$S \to N^1 \to N^3 \to N^4 \to D$	4	4
	P5	$S \to N^1 \to N^2 \to N^4 \to D$	4	4