




On the shortest α -reliable path problem

David Corredor-Montenegro¹ · Nicolás Cabrera¹ · Raha Akhavan-Tabatabaei² · Andrés L. Medaglia¹ 

Received: 16 May 2020 / Accepted: 5 January 2021 / Published online: 8 March 2021
© Sociedad de Estadística e Investigación Operativa 2021

Abstract

In this variant of the constrained shortest path problem, the time of traversing an arc is given by a non-negative continuous random variable. The problem is to find a minimum cost path from an origin to a destination, ensuring that the probability of reaching the destination within a time limit meets a certain reliability threshold. To solve this problem, we extend the pulse algorithm, a solution framework for shortest path problems with side constraints. To allow arbitrary non-negative continuous travel-time distributions, we model the random variables of the travel times using Phase-type distributions and Monte Carlo simulation. We conducted a set of experiments over small- and medium-size stochastic transportation networks with and without spatially-correlated travel times. As an alternative to handling correlations, we present a scenario-based approach in which the distributions of the arc travel times are conditioned to a given scenario (e.g., variable weather conditions). Our methodology and experiments highlight the relevance of considering on-time arrival probabilities and correlations when solving shortest path problems over stochastic transportation networks.

Keywords Constrained shortest path problem · Pulse algorithm · Stochastic shortest path · Phase-type distributions · Spatial correlation · Chance constraints

Mathematical Subject Classification 90-08 · 90B15 · 90B06 · 90C35

✉ Andrés L. Medaglia
amedagli@uniandes.edu.co

David Corredor-Montenegro
d.corredor@uniandes.edu.co

Nicolás Cabrera
n.cabrera10@uniandes.edu.co

Raha Akhavan-Tabatabaei
akhavan@sabanciuniv.edu

¹ Centro para la Optimización y Probabilidad Aplicada (COPA), Departamento de Ingeniería Industrial, Universidad de los Andes, Bogotá, Colombia

² Business School, Sabanci University, Istanbul, Turkey

1 Introduction

Transportation networks are subject to uncertainty caused by random events such as changing weather conditions, demand fluctuations, traffic delay at intersections, signal failures, mixed traffic flow, collisions, vehicle breakdowns, or even natural disasters. Consequently, travel times in transportation networks are inherently uncertain and neglecting their stochastic nature may lead to poor routing decisions.

In the case of finding shortest paths in a network, several efficient algorithms exist for graphs with deterministic parameters (e.g., cost or time) (Dijkstra 1959; Gallo and Pallottino 1988; Ahuja et al. 1993). However, leveraging on such algorithms to identify the path with the least expected travel time on a stochastic network, may not necessarily lead to a reliable solution in terms of travel times (Zeng et al. 2015). Hence, several authors have modeled travel times as random variables to capture the underlying uncertainty on stochastic transportation networks.

In the literature of stochastic shortest path problems, the pioneering work by Frank (1969) estimated the probability distribution of the shortest path and compared paths pairwise according to the probability that the shortest path is less than a given threshold. This work assumes that the probabilistic information is known and continuous. Later, Mirchandani (1976) proposed a recursive algorithm to solve a discrete version of Frank's problem, which entailed enumerating paths and travel times. Sigal et al. (1980) derived the probability that a given path is shorter than all other paths in a network through the concept of path optimality index. Sivakumar and Batta (1994) solved the problem of finding a path with the least expected travel cost, subject to not exceeding a given threshold on the variance of the travel cost. After linearizing this constrained shortest path, they solved the problem with Lagrangean relaxation on a 50-node network. An alternative bi-objective approach by Sen et al. (2001), simultaneously minimizes the travel time's mean and variance for the whole path. Their approach requires to solve a series of relaxed 0–1 quadratic programming models. They solved instances on sparse networks with less than 70 nodes. The least expected travel time (LET) problem on stochastic and time-varying networks (Hall 1986; Miller-Hooks and Mahmassani 1998) has led to a significant stream of research in stochastic networks (Miller-Hooks and Mahmassani 2000; Miller-Hooks 2001; Miller-Hooks and Mahmassani 2003; Prakash 2018). Although the LET problem considers uncertainty, it often falls short in coping with the risk-averse behavior of travelers. Yang et al. (2013) proposed a two-stage stochastic programming model that considers discrete random scenario-based travel times. Wang et al. (2016) formulated the stochastic constrained shortest path problem as a binary programming model that finds a path with the least expected travel time. They model travel times using discrete random variables and allow a correlation structure. They proposed a Lagrangean relaxation-based algorithm to solve this constrained shortest path problem.

In the context of stochastic transportation networks, Chen and Ji (2005) proposed the concept of an α -reliable path, with the premise that it is of utmost

importance for travelers to reach their destination on time. The problem is to find a path with the minimum travel time budget, ensuring a given on-time arrival probability α , that is, the probability of not exceeding the time budget is at least α . They formulated the α -reliable path problem as a chance-constrained optimization model and solved it with a simulation-based genetic algorithm. The seminal work by Chen and Ji (2005) led to a fruitful stream of research on α -reliable paths. Ji et al. (2011) extended the α -reliable path to a multiobjective version to accommodate multiple confidence requirements (e.g., the probability of on-time arrival and average time) and incorporated travel time (spatial) correlations. They solved the underlying multi-objective chance-constrained optimization model with a multi-objective simulation-based genetic algorithm. Nie and Wu (2009) developed a label-correcting algorithm to solve the most reliable path problem by generating non-dominated paths under first-order stochastic dominance. Chen et al. (2012) solved the reliable shortest path where travel times are spatially correlated using a multi-criteria A* algorithm. Zeng et al. (2015) solved an α -reliable path in a stochastic network with spatially correlated travel times. To make it computationally tractable, the path travel time distribution was approximated by a normal distribution. They solved the problem using Lagrangean relaxation. Recently, Shen et al. (2019) proposed a novel application in the context of electric vehicles, where it is necessary to find reliable and energy-efficient paths in stochastic traffic networks with spatially-correlated travel times. They proposed a biobjective optimization approach capable of unveiling non-dominated solutions.

Several stochastic variants of shortest path problems can be seen as special cases of constrained shortest paths. Two prominent examples are the variance-constrained shortest path problem by Sivakumar and Batta (1994) and the stochastic constrained shortest path problem by Wang et al. (2016). Many approaches have been proposed in the literature to tackle the constrained shortest path (CSP) problem, including dynamic programming based labeling algorithms (Dumitrescu and Boland 2003; Joksch 1966; Thomas et al. 2019) and path ranking approaches (Handler and Zang 1980; Santos et al. 2007; Sedeño-Noda and Alonso-Rodríguez 2015). A competitive approach for the CSP was proposed by Lozano and Medaglia (2013) and coined with the term *pulse algorithm*. This exact algorithm uses depth-first search combined with effective pruning strategies to make an implicit exploration of the solution space of the CSP. In addition to being one of the state-of-the-art algorithms for the CSP, efficiently solving real-road networks with up to 6 million nodes and 15 million arcs in Cabrera et al. (2020), the pulse algorithm has been successfully extended to solve other hard shortest path variants. The same principles have been applied to solve the elementary shortest path problem with resource constraints (Lozano et al. 2016; Li and Han 2019), the biobjective shortest path problem (Duque et al. 2015), the weight constrained shortest path problem with replenishment (Bolívar et al. 2014), the orienteering problem with time windows (Duque et al. 2014), and the scenario-based robust shortest path problem (Duque and Medaglia 2019), among other shortest path variants.

The aim of this paper is to contribute to the body of knowledge in the domain of stochastic shortest path problems in transportation networks. First, we introduce the shortest α -reliable path (S- α RP) as the problem of finding a minimum-cost

path ensuring a given on-time arrival probability α for a given travel time budget. Second, we consider travel times as non-negative continuous random variables, without making strong distribution assumptions (i.e., normality or additivity). To overcome the usual difficulties found in the literature, we propose to model travel times using phase-type (PH) distributions. Using this family of distributions, we can closely approximate any positive, continuous distribution and compute the on-time arrival probability with arbitrary precision. Third, we present a Monte Carlo simulation approach to compute the on-time arrival probability with spatially-correlated travel times. Our approach throughout the paper is adaptable to many situations given the availability of travel time data from sensors or mobile devices, to which we can fit the flexible PH distributions. Fourth, we introduce an alternative scenario-based approach where it is possible to model some situations where correlated travel times arise as the product of external factors (e.g., weather) influencing the whole state of the network. Finally, by modeling the problem as a constrained shortest path problem, we are able to extend the pulse algorithm (Lozano and Medaglia 2013) to solve the S- α RP problem efficiently.

The rest of the paper is organized as follows. Section 2 formally defines the S- α RP. Section 3 describes the proposed algorithm that solves the S- α RP and shows in detail its core components. Section 4 describes the two proposed ways to model the stochastic travel times and shows how to compute estimations of the on-time arrival probability. Section 5 presents the computational experiments of each of the versions of the proposed algorithm over small- and medium-size stochastic transportation networks with and without spatially-correlated travel times. This section also presents the results of the scenario-based approach to handle correlations. Finally, Sect. 6 concludes the paper and outlines future work.

2 Problem definition

The shortest α -reliable path problem is defined over a directed graph $\mathcal{G} = (\mathcal{N}, \mathcal{A})$, where $\mathcal{N} = \{v_1, \dots, v_i, \dots, v_n\}$ is the set of nodes, and $\mathcal{A} \subseteq \{(i, j) | v_i \in \mathcal{N}, v_j \in \mathcal{N}, i \neq j\}$ is the set of arcs. Each arc $(i, j) \in \mathcal{A}$ has a cost $c_{i,j}$ (e.g., the expected travel time, distance, or fuel cost) and a time represented by the random variable $\tilde{t}_{i,j}$. A path $\mathcal{P}_{i,j}$ from node v_i to v_j is described as a sequence of nodes $\{v_{(1)}, \dots, v_{(k)}, \dots, v_{(r)}\}$ where $v_{(1)} = v_i$, $v_{(r)} = v_j$, $v_{(k)}$ represents the node at the k -th position of the sequence and $(v_{(k)}, v_{(k+1)}) \in \mathcal{A}$ for $1 \leq k \leq r - 1$. With a slight abuse of notation, we also refer to a path $\mathcal{P} = \{\dots, (i, j)_{(k)}, \dots\}$ as an ordered sequence of arcs, where arc $(i, j)_{(k)} \in \mathcal{A}$ is the k th arc in the sequence. In general, the set of all possible paths from node $v_i \in \mathcal{N}$ to node $v_j \in \mathcal{N}$ is denoted by $\Omega_{i,j}$.

S- α RP consists of finding a minimum cost path from a start node $v_s \in \mathcal{N}$ to an end node $v_e \in \mathcal{N}$, while ensuring a given on-time arrival probability α , that is, the probability of not exceeding the maximum time budget T is at least α . A mathematical formulation of the S- α RP is stated as follows:

$$\begin{aligned} & \min_{\mathcal{P} \in \Omega_{s,e}} c(\mathcal{P}) \\ & \text{s.t. } \mathbb{P}(\tilde{t}(\mathcal{P}) \leq T) \geq \alpha, \end{aligned} \tag{S- α RP}$$

where $c(\mathcal{P}) \triangleq \sum_{(i,j) \in \mathcal{P}} c_{i,j}$ and $\tilde{t}(\mathcal{P}) \triangleq \sum_{(i,j) \in \mathcal{P}} \tilde{t}_{i,j}$ represent the cost and the random variable of the traversing time of path \mathcal{P} , respectively. We call a path $\mathcal{P} \in \Omega_{s,e}$ a *complete path* from the start node $v_s \in \mathcal{N}$ to the end node $v_e \in \mathcal{N}$.

3 Methodology

In this section, we present an overview of the original pulse algorithm (PA) by Lozano and Medaglia (2013), followed by a detailed explanation of the proposed pruning strategies and extensions required to solve the S- α RP.

3.1 Overview of the pulse algorithm

The pulse algorithm (PA) explores the network by propagating recursively a partial path (*pulse*) from the start node $v_s \in \mathcal{N}$. This pulse, traverses the network, storing crucial information, like the nodes in path $\mathcal{P}_{s,i}$, the cost, and the resource consumption. Once a pulse reaches the target node $v_e \in \mathcal{N}$, the PA tries to update an upper bound on the minimal cost and backtracks to continue the recursive search through the network. If nothing prevents the pulse propagation, the PA enumerates all paths from v_s to v_e ensuring that the optimal path \mathcal{P}^* is found.

Considering that the number of paths grows exponentially as the size of the network grows, what makes the PA efficient is how it uses the pulse information to prevent it from propagating when there is enough evidence that the partial path will not lead to an improved or feasible solution. For this matter, several pruning strategies have been developed. The three core strategies proposed by Lozano and Medaglia (2013) are pruning by *infeasibility* (see Sect. 3.2.1), *bounds* (see Sect. 3.2.2) and *dominance* (see Sect. 3.2.3). The pruning by infeasibility strategy stops propagating a pulse as soon as it is known that it will not meet the side constraint(s); the pruning by bounds strategy prunes the pulse as soon as it is known that it will not improve the current best solution; and finally, dominance relationships help prune dominated partial paths. The earlier the pulses are pruned, the fewer paths are explored, thus improving the algorithm’s performance.

3.2 Pulse extensions for the S- α RP

The input for the PA for the S- α RP are the graph \mathcal{G} , a start node v_s , an end node v_e , the maximum time T , and the on-time arrival probability (reliability threshold) α . The output of PA is a minimum cost reliable path, namely, a path that satisfies the on-time arrival probabilistic constraint $\mathbb{P}(\tilde{t}(\mathcal{P}) \leq T) \geq \alpha$, if such path exists. For the S- α RP, a pulse travels through the network with the cumulative

cost, $c(\mathcal{P})$, and the random cumulative time, $\tilde{t}(\mathcal{P})$. Algorithm 1 outlines the PA for the S- α RP. Lines 1–3 initialize \mathcal{P} , $c(\mathcal{P})$ and $\tilde{t}(\mathcal{P})$. Line 4 calculates shortest paths from any node v_i to the end node v_e . This preprocessing step also uses properties of the random variables to calculate free-flow travel times from each node $v_i \in \mathcal{N}$ in the network to the end node v_e . Line 5 invokes the recursive pulse function (Algorithm 2) from the start node v_s . Finally, line 6 returns an optimal path found in the recursion.

Algorithm 1 Pulse Algorithm for the S- α RP

Require: \mathcal{G} , directed graph; v_s , start node; v_e , end node; T , maximum time; α , reliability threshold.

Ensure: \mathcal{P}^* , optimal path.

- 1: $\mathcal{P}^* \leftarrow \emptyset$
 - 2: $c(\mathcal{P}) \leftarrow 0$
 - 3: $\tilde{t}(\mathcal{P}) \leftarrow 0$
 - 4: **preprocess**(\mathcal{G})
 - 5: **pulse**($v_s, c(\mathcal{P}), \tilde{t}(\mathcal{P}), \mathcal{P}$)
 - 6: **return** \mathcal{P}^*
-

Algorithm 2 shows the recursive function pulse for the S- α RP. Lines 1–3 perform Boolean function checks that return true if the pulse is pruned (false, otherwise). Line 4 updates the partial path. Lines 5–8 propagate a pulse to all nodes in the forward star of v_i , i.e., $v_j \in \Gamma^+(v_i)$, where $\Gamma^+(v_i) = \{v_j \in \mathcal{N} | (i, j) \in \mathcal{A}\}$. For each node $v_j \in \Gamma^+(v_i)$, lines 6 and 7 update the partial cost and the random variable of the time of the partial path $\mathcal{P}_{s,j}$, and finally, line 8 invokes again the pulse function. Every time the pulse function is invoked on the final node v_e , the information of the best path (incumbent) is updated, the pulse is no longer propagated, and backtracking starts. The following sections describe in detail the pruning strategies for the S- α RP.

Algorithm 2 Recursive function pulse

Require: v_i , current node; $c(\mathcal{P})$, cumulative cost; $\tilde{t}(\mathcal{P})$, cumulative time; \mathcal{P} , partial path.

- 1: **if** \neg **check_feasibility**($v_i, \tilde{t}(\mathcal{P})$) **then** \triangleright see §3.2.1
 - 2: **if** \neg **check_bounds**($v_i, c(\mathcal{P})$) **then** \triangleright see §3.2.2
 - 3: **if** \neg **check_dominance**($v_i, c(\mathcal{P}), \tilde{t}(\mathcal{P})$) **then** \triangleright see §3.2.3
 - 4: $\mathcal{P} \leftarrow \mathcal{P} \cup \{v_i\}$
 - 5: **for** $v_j \in \Gamma^+(v_i)$ **do**
 - 6: $c(\mathcal{P}') \leftarrow c(\mathcal{P}) + c_{i,j}$
 - 7: $\tilde{t}(\mathcal{P}') \leftarrow \tilde{t}(\mathcal{P}) + \tilde{t}_{i,j}$
 - 8: **pulse**($v_j, c(\mathcal{P}'), \tilde{t}(\mathcal{P}'), \mathcal{P}'$)
 - 9: **end for**
 - 10: **end if**
 - 11: **end if**
 - 12: **end if**
-

3.2.1 Infeasibility pruning

The idea behind this pruning strategy is to discard paths at an early stage, as soon as it is known that the traveling pulse will not be able to reach the final node v_e meeting the chance constraint. Thus, if $\mathbb{P}(\tilde{t}(\mathcal{P}_{s,i}) \leq T) < \alpha$ we can safely say that any path from v_s to v_e that contains the partial path $\mathcal{P}_{s,i}$ will not meet the chance constraint, that is, it will not arrive to the node v_e with the required on-time arrival probability.

We can further strengthen this strategy, based on some properties of the underlying random variables. Since we are modeling travel times, we assume that each random variable has a non-negative lower bound. This lower bound corresponds to the travel time at free-flow speed (Gómez et al. 2016). Accordingly, we decompose each $\tilde{t}_{i,j}$ into a deterministic part accounting for this free-flow speed time, denoted by $t_{i,j}$, and a stochastic component that considers the variability of each arc travel time, denoted by $\hat{t}_{i,j}$. Figure 1 shows the probability density functions, $f_{\tilde{t}_{i,j}}(t)$ and $f_{\hat{t}_{i,j}}(t)$, respectively. Accordingly, we have that $\tilde{t}_{i,j} = t_{i,j} + \hat{t}_{i,j}$, and that $t_{i,j}, \hat{t}_{i,j} \geq 0$.

Considering the deterministic part of the arc travel times, the minimum time consumption $t(i)$ from any node v_i to the final node v_e under the best possible scenario can be computed by minimum-time paths as in Lozano and Medaglia (2013). This best possible scenario is the one in which every arc traversal is done at free-flow speed, namely, a road network in ideal conditions. This $t(i)$ corresponds to the time of the minimum-time path $\mathcal{P}_{i,e}^t$ from node $v_i \in \mathcal{N}$ to the end node $v_e \in \mathcal{N}$, under the best possible scenario. The procedure required to find all the $t(i)$ values is pre-computed at the initialization of the PA.

With the $t(i)$ for each node $v_i \in \mathcal{N}$, we can improve our infeasibility pruning strategy by computing the probability of arriving on time to the end node v_e , under the best possible scenario for path $\mathcal{P}_{i,e}$. If under this scenario the chance constraint is not met, it is not possible for the path to arrive on time to the end node v_e , under other less favorable scenarios. The probability of arriving on time under the best possible scenario denoted by $\pi(\mathcal{P}_{s,i})$, is computed as follows:

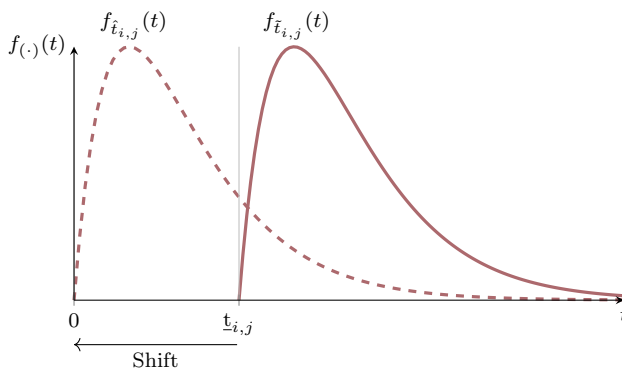


Fig. 1 Shifted travel time distributions

$$\begin{aligned} \pi(\mathcal{P}_{s,i}) &= \mathbb{P}(\tilde{t}(\mathcal{P}_{s,i} \cup \mathcal{P}_{i,e}^t) \leq T | \tilde{t}(\mathcal{P}_{i,e}^t) = t(i)) \\ &= \mathbb{P}(\tilde{t}(\mathcal{P}_{s,i}) + t(i) \leq T) \\ &= \mathbb{P}(\tilde{t}(\mathcal{P}_{s,i}) \leq T - t(i)). \end{aligned}$$

Let $T(i) \triangleq T - t(i)$ be the maximum allowed time consumption for a partial path up to node v_i . If for a partial path $\mathcal{P}_{s,i}$ arriving to node v_i , the probability of arriving on time to the end node v_e under the best scenario does not meet the reliability threshold α , then the partial path $\mathcal{P}_{s,i}$ is already infeasible and it is not worth keep propagating that pulse. In other words if $\pi(\mathcal{P}_{s,i}) = \mathbb{P}(\tilde{t}(\mathcal{P}_{s,i}) \leq T(i)) \leq \alpha$, then the pulse represented by the partial path $\mathcal{P}_{s,i}$ can be safely pruned.

3.2.2 Bounds pruning

This pruning strategy remains unchanged from the original bounds pruning strategy presented by Lozano and Medaglia (2013), although for the sake of completeness we briefly describe it here. Every time a feasible solution is found (i.e., every time a pulse reaches the final node), a global primal bound is updated to keep track of the minimum cost. We use this primal bound, denoted by \bar{c} , to prune partial paths that promise no improvement. Similarly to the minimum time consumption $t(i)$, we denote a bound $c(i)$ with the minimum-cost path $\mathcal{P}_{i,e}^c$ from node v_i to the end node v_e . As these paths $\mathcal{P}_{i,e}^c$ ignore the time consumption, the value $c(i)$ is a lower bound on the minimum cost subject to the chance constraint. In this sense, given a partial path $\mathcal{P}_{s,i}$ from v_s to v_i , we can safely prune the pulse if $c(\mathcal{P}_{s,i}) + c(i) > \bar{c}$. This means that as soon as we know that a partial path does not promise an improvement on the global cost, it is not worth keep propagating that pulse.

3.2.3 Dominance pruning

During the recursive search, a node v_i could be reached more than once by different pulses. By defining dominance relations between paths, we can compare them and decide which paths are worth extending and which are better not.

Definition 1 (Dominance) Let $\mathcal{P}_{s,i}^1, \mathcal{P}_{s,i}^2 \in \Omega_{s,i}$ be partial paths from the start node v_s to node $v_i \in \mathcal{N}$. We say that partial path $\mathcal{P}_{s,i}^1$ strongly dominates $\mathcal{P}_{s,i}^2$ if

$$c(\mathcal{P}_{s,i}^1) < c(\mathcal{P}_{s,i}^2) \quad \text{and} \quad \pi(\mathcal{P}_{s,i}^1) > \pi(\mathcal{P}_{s,i}^2).$$

In addition, we say that partial path $\mathcal{P}_{s,i}^1$ weakly dominates $\mathcal{P}_{s,i}^2$ if

$$\begin{aligned} &c(\mathcal{P}_{s,i}^1) = c(\mathcal{P}_{s,i}^2) \quad \text{and} \quad \pi(\mathcal{P}_{s,i}^1) > \pi(\mathcal{P}_{s,i}^2) \\ \text{or} \quad &c(\mathcal{P}_{s,i}^1) < c(\mathcal{P}_{s,i}^2) \quad \text{and} \quad \pi(\mathcal{P}_{s,i}^1) = \pi(\mathcal{P}_{s,i}^2). \end{aligned}$$

Given these dominance relations, we use a *limited* set of labels to check if an incoming pulse to node $v_i \in \mathcal{N}$ is dominated. We define a limited set of labels $L(v_i) = \{(c_l, \pi_l) | l = 1, \dots, R\}$, where c_l and π_l are the cost and probability (defined in Sect. 2), respectively; and R denotes the total number of labels (i.e., memory size). More importantly, the fact that not all labels are stored does not preclude us of finding the optimal solution. Although it is possible that some dominated paths may inefficiently pass through node v_i and will reach the end node v_e , this action will never prune the optimal solution.

4 Probability estimation

The proposed PA for the S- α RP relies on the calculation of the on-time arrival probability (reliability) of each partial path $\mathcal{P}_{s,i}$. This task requires computing the convolution of travel time random variables along the path. Depending on the probability distributions assumed for the random variables, this can be a trivial (e.g., normal distributions) or a complex task (e.g., lognormal distributions).

Gómez et al. (2016) discuss the difficulty of choosing a unique family of distributions with additive properties to model travel times. They propose the application of phase-type (PH) distributions as a versatile family that could provide many appropriate shapes of the probability density function. Furthermore, the additive property holds for this family of distributions, making it a good fit to model stochastic travel times in shortest path problems.

In this section, we present two distinct approaches for the estimation of the reliability of a path. Following Gómez et al. (2016), our first approach is to model random travel times with PH distributions, taking advantage of their properties to model the travel times for each arc $(i,j) \in \mathcal{A}$, a crucial decision when implementing the PA for solving the S- α RP. Under the assumption of independence, this family of distributions allows us to approximate any positive continuous distribution with precision, compute convolutions exactly, and express in closed form its cumulative distribution function, and thus the reliability (in terms of on-time arrival probability) for any path.

In our context, the application of PH distributions involves an initial fitting phase and assuming independence between the travel times of adjacent arcs. To relax the assumption of independence, our second approach employs Monte Carlo simulation to estimate the on-time arrival probabilities under spatially-correlated networks.

We begin this section by introducing the PH family of distributions and describing our first approach to model travel times via the PH distribution. Then, we present our Monte Carlo approach and in the next section we demonstrate their respective performance on a series of computational experiments.

4.1 Phase-type distributions

This section introduces PH distributions and states the main properties that support the PA for the S- α RP. Latouche and Ramaswami (1999) define the PH family of

distributions as the distribution of the time until absorption of a continuous time Markov chain (CTMC) with one absorbing state and all others transient. Consider a continuous time Markov process on the states $\{0, 1, \dots, n\}$ with initial probability vector $(\tau_0, \boldsymbol{\tau})$ and infinitesimal generator:

$$Q = \begin{bmatrix} 0 & \mathbf{0} \\ \mathbf{z} & \mathbf{Z} \end{bmatrix},$$

where $\boldsymbol{\tau}$ is a row vector of size n , \mathbf{Z} is an $n \times n$ matrix, and \mathbf{z} is a column vector of size n . Since Q is a generator matrix of a CTMC, we must have that $Z_{ii} < 0$, $Z_{ij} \geq 0$, $z_i \geq 0$ for $1 \leq i \neq j \leq n$, and $\mathbf{Z}\mathbf{1} + \mathbf{t} = \mathbf{0}$, where $\mathbf{1}$ is a column vector of ones. We also have that $\tau_0 + \mathbf{1}\boldsymbol{\tau} = 1$ and that τ_0 represents the probability of beginning in the absorbing state (Latouche and Ramaswami 1999). In this way, any PH distribution is completely determined by the parameters $(\boldsymbol{\tau}, \mathbf{Z})$. Henceforth, a random variable X that follows a PH distribution will be denoted as $X \sim PH(\boldsymbol{\tau}, \mathbf{Z})$.

Under this approach, modeling the stochastic travel times is based on two assumptions. First, we assume that all travel times can be well approximated by PH distributions, and second, we assume that all the \tilde{t}_{ij} are independent random variables. Under these assumptions, we benefit from the following three properties that allow us to estimate and express in closed form the on-time arrival probabilities.

First, PH distributions are dense in the set of continuous density functions with support on $[0, \infty)$. This means that there exists a PH distribution arbitrarily close to any positive distribution, as discussed in chapter 2.7 of Latouche and Ramaswami (1999). Accordingly, PH distributions are a versatile tool for modeling stochastic travel times, considering that travel times are always positive (Gómez et al. 2016). Finding those PH distributions, based on data describing arc travel times, is a process known as distribution fitting. Many efficient algorithms exist for fitting PH distributions to data. These algorithms estimate the parameters either by matching moments or by solving a nonlinear problem for maximum likelihood or minimum distance estimation (Bobbio et al. 2005; Asmussen et al. 1996; Thummler et al. 2006).

Second, the sum of finite PH distributed random variables is again a PH distribution (see Theorem 2.6.1 in Latouche and Ramaswami (1999)). Let $\tilde{t}_{i,j}$ and $\tilde{t}_{k,l}$ be two independent random variables modelling the random travel times of arcs $(i, j) \in \mathcal{A}$ and $(k, l) \in \mathcal{A}$, respectively. Let us assume that $\tilde{t}_{i,j} \sim PH(\boldsymbol{\tau}_{i,j}, \mathbf{Z}_{i,j})$ with $n_{i,j}$ phases, and $\tilde{t}_{k,l} \sim PH(\boldsymbol{\tau}_{k,l}, \mathbf{Z}_{k,l})$ with $n_{k,l}$ phases. Then, their sum $\tilde{t}_{i,j} + \tilde{t}_{k,l}$ follows a $PH(\boldsymbol{\tau}_{i,k}, \mathbf{Z}_{i,k})$ with $n_{i,j} + n_{k,l}$ phases, where

$$\boldsymbol{\tau}_{i,k} = [\boldsymbol{\tau}_{i,j}, \tau_{i,j_0} \cdot \boldsymbol{\tau}_{k,l}] \tag{4.1}$$

and

$$\mathbf{Z}_{i,k} = \begin{bmatrix} \mathbf{Z}_{i,j} & \mathbf{z}_{i,j} \cdot \boldsymbol{\tau}_{k,l} \\ \mathbf{0} & \mathbf{Z}_{k,l} \end{bmatrix}. \tag{4.2}$$

Due to the closure property of the PH distributions, the random variable representing the travel time of every path in the graph \mathcal{G} , follows a PH distribution with

parameters $(\tau_{\mathcal{P}}, \mathbf{Z}_{\mathcal{P}})$. Note that $\tau_{\mathcal{P}}$ and $\mathbf{Z}_{\mathcal{P}}$ can be computed inductively from $\tau_{i,j}$ and $\mathbf{Z}_{i,j}$ for $(i,j) \in \mathcal{P}$ with Eqs. (4.1) and (4.2).

Third, and last, the cumulative distribution function of a PH can be expressed in closed form in terms of τ and \mathbf{Z} (see Theorem 2.4.1 in Latouche and Ramaswami (1999)). So, if $\tilde{t}_{\mathcal{P}} \sim PH(\tau_{\mathcal{P}}, \mathbf{Z}_{\mathcal{P}})$, then its distribution function is given by

$$\mathbb{P}(t_{\mathcal{P}} \leq x) = F(x) = 1 - \tau_{\mathcal{P}} \cdot \exp(\mathbf{Z}_{\mathcal{P}} \cdot x) \cdot \mathbf{1}, \quad \text{for } x \geq 0, \tag{4.3}$$

where the matrix exponential is defined by

$$\exp(A) = \sum_{n=0}^{\infty} \frac{1}{n!} A^n. \tag{4.4}$$

By modeling the stochastic travel times as independent PH random variables, the on-time arrival probability to the end node v_e given the free-flow speed scenario, namely $\pi(\mathcal{P}_{s,i})$, is given by the following closed form expression:

$$\pi(\mathcal{P}_{s,i}) = \mathbb{P}(\tilde{t}(\mathcal{P}_{s,i}) < T(i)) = 1 - \tau_{\mathcal{P}_{s,i}} \cdot \exp(\mathbf{Z}_{\mathcal{P}_{s,i}}) \cdot \mathbf{1}. \tag{4.5}$$

4.2 Monte Carlo simulation

Aside from using PH distributions we propose an alternative way of estimating the probabilities $\pi(\mathcal{P}_{s,i})$ for each partial path within the PA for the S- α RP. In contrast to the PH distribution approach, this alternative does not assume any family of distributions over the arc travel times in the network, nor it assumes independence. This method uses for each arc in the network the distribution that best fits its data and on-time arrival probabilities are estimated via Monte Carlo simulation.

In general, Monte Carlo simulation estimates expectations of the form $\mathbb{E}[g(\mathbf{Y})]$, where \mathbf{Y} is a random vector (variable) of dimension \mathbb{R}^m , and $g : \mathbb{R}^m \rightarrow \mathbb{R}$ is an arbitrary function. So given a sample $\mathbf{Y}_1, \dots, \mathbf{Y}_n$ of n independent and identically distributed (i.i.d) realizations of the random vector \mathbf{Y} , a Monte Carlo estimator of $\mathbb{E}[g(\mathbf{Y})]$ is given by the sample mean $\hat{g}(\mathbf{Y}) = \frac{1}{n} \sum_{i=1}^n g(\mathbf{Y}_i)$.

To estimate the on-time arrival probability $\pi(\mathcal{P})$ of the incoming partial path $\mathcal{P} \in \Omega_{s,v}$ to node $v \in \mathcal{N}$, we must formally define its Monte Carlo estimator. By definition, we have that for each partial path \mathcal{P} in the graph, $\tilde{t}(\mathcal{P}) = \sum_{(i,j) \in \mathcal{P}} \tilde{t}_{i,j}$. We let

$$\mathbf{t}_{\mathcal{P}} = \begin{pmatrix} \tilde{t}_{(i,j)_{(1)}} \\ \vdots \\ \tilde{t}_{(i,j)_{(k)}} \\ \vdots \\ \tilde{t}_{(i,j)_{(|\mathcal{P}|)}} \end{pmatrix}$$

be a random vector, where $\tilde{t}_{(i,j)_{(k)}}$ represents the random travel time of the k -th arc of the path. We define the indicator function of the event $\{\mathbf{1}^T \mathbf{t}_{\mathcal{P}} \leq T(v)\}$ as

$$I_{\{\mathbf{1}^T \mathbf{t}_{\mathcal{P}} \leq T(v)\}} = \begin{cases} 1 & \text{if } \mathbf{1}^T \mathbf{t}_{\mathcal{P}} \leq T(v) \\ 0 & \text{otherwise} \end{cases}$$

and we have that

$$\pi(\mathcal{P}) = \mathbb{P}(\tilde{t}(\mathcal{P}) \leq T(v)) = \mathbb{E} \left[I_{\{\mathbf{1}^T \mathbf{t}_{\mathcal{P}} \leq T(v)\}} \right].$$

Let $\mathbf{t}_{\mathcal{P}}^{(1)}, \dots, \mathbf{t}_{\mathcal{P}}^{(\ell)}, \dots, \mathbf{t}_{\mathcal{P}}^{(n)}$ be n i.i.d realizations of the random vector $\mathbf{t}_{\mathcal{P}}$, a Monte Carlo estimator for $\pi(\mathcal{P})$ is given by

$$\hat{\pi}(\mathcal{P}) = \frac{1}{n} \sum_{\ell=1}^n I_{\{\mathbf{1}^T \mathbf{t}_{\mathcal{P}}^{(\ell)} \leq T(v)\}}.$$

Note that $\left\{ \mathbf{t}_{\mathcal{P}}^{(\ell)} \right\}_{\ell=1, \dots, n}$ being i.i.d does not mean that the $(\tilde{t}_{(i,j)})_{(i,j) \in \mathcal{P}}$ are independent. To induce a correlation structure in the estimation of the probability $\pi(\mathcal{P})$, we must generate the samples of $\mathbf{t}_{\mathcal{P}}$ with target correlations. Therefore, our goal is to generate random vectors with the following two properties:

1. Let us denote by F_k the cumulative distribution function of $\tilde{t}_{(i,j)(k)}$, the random time of the k -th arc of path \mathcal{P} . Then, $\left\{ \tilde{t}_{(i,j)(k)}^{(\ell)} \right\}_{\ell=1, \dots, n} \sim F_k$ for $k = 1, \dots, |\mathcal{P}|$.
2. $\text{Cov}[\mathbf{t}_{\mathcal{P}}] = \Sigma_{\mathbf{t}_{\mathcal{P}}}$, where $\Sigma_{\mathbf{t}_{\mathcal{P}}}$ corresponds to the target covariance matrix of $\mathbf{t}_{\mathcal{P}}$.

There are several methods that allow us to generate correlated random vectors. Haas (1999) proposed a procedure based on copulas, in which the copula and its parameters have to be specified as input to the random number generation procedure. Finding the appropriate copula for any given correlation metric may be challenging, thus, the majority of methods focus on transforming multivariate normal distribution into a multivariate distribution with target marginals and correlations. Li and Hammond (1975) proposed an analytical method based on this principle, but their procedure leads to the numerical solution of double-integral equations that might become computationally intensive and unstable given a certain degree of accuracy. To overcome this practical limitation, van der Geest (1998) developed an algorithm to stabilize and increase the accuracy of Li and Hammond’s method while Lurie and Goldberg (1998) presented a modified version in which a non linear optimization procedure minimizes the distance between the achieved and target correlation matrix. Cario and Nelson (1997) proposed NORTA (normal-to-anything), an approach in which a multivariate normal distribution is transformed into any multivariate distribution with a target correlation matrix.

We use the NORTA method to generate the correlated random vectors by presenting $\mathbf{t}_{\mathcal{P}}$ as a transformation of a $|\mathcal{P}|$ -dimensional, standard multivariate normal vector $\mathcal{Z} = (\mathcal{Z}_1, \dots, \mathcal{Z}_{|\mathcal{P}|})$ with covariance matrix $\Sigma_{\mathcal{Z}}$ as follows:

$$t_{\mathcal{P}} = \begin{pmatrix} \tilde{t}_{(i,j)_{(1)}} \\ \vdots \\ \tilde{t}_{(i,j)_{(k)}} \\ \vdots \\ \tilde{t}_{(i,j)_{(|\mathcal{P}|)}} \end{pmatrix} = \begin{pmatrix} F_1^{-1}(\Phi(\mathcal{Z}_1)) \\ \vdots \\ F_k^{-1}(\Phi(\mathcal{Z}_k)) \\ \vdots \\ F_{|\mathcal{P}|}^{-1}(\Phi(\mathcal{Z}_{|\mathcal{P}|})) \end{pmatrix}.$$

As stated by Cario and Nelson (1997), the difficulty of this procedure is to select an accurate correlation matrix $\Sigma_{\mathcal{Z}}$ that produces the target correlation matrix $\Sigma_{t_{\mathcal{P}}}$.

5 Computational experiments

We begin this section by describing the test instances used for our computational analyses. Then, we proceed to test the proposed algorithm in its two variants. First, we model the stochastic travel times as flexible (yet independent) PH random variables (PA-PH) and use their properties to compute the on-time arrival probabilities. Second, we model the stochastic travel times with arbitrary distributions and estimate the probabilities via Monte Carlo simulation (PA-MC). In addition, to illustrate the effects of spatial correlation when solving the S- α RP, we present an experiment over the Sioux Falls network and compare the results with and without spatial correlations using PA-MC. Finally, we present experiments in which the distributions of the arc travel times are conditioned to a given scenario (e.g., variable weather conditions), in order to further illustrate an alternative to handling correlations.

5.1 Instances and implementation

To test our proposed algorithm, we adapted two networks from a repository (Stabler 2016) widely used in the literature in the context of stochastic transportation problems (Chen and Nie 2015; Li et al. 2010; Nie and Wu 2009). For these networks, the repository provides the cost, free-flow time, the expected value of the traversing time, the average flow (in terms of vehicles per hour), and the capacity (in terms of vehicles per hour) for each arc. Since the repository does not contain data samples, we adapted these networks to the stochastic context assuming that all travel times are lognormally distributed. We fixed the mean of the lognormal distribution to the value given by the repository and determined its standard deviation as a function of the coefficient of variation (CV) of the shifted distribution. More specifically, for each arc $(i, j) \in \mathcal{A}$ and given a CV the standard deviation is assumed to be $\sigma_{i,j} = CV \cdot (\mathbb{E}[\tilde{t}_{(i,j)}] - t_{i,j})$. Note that the CV will be treated as a parameter for the experiments, and it will be the same for all travel times in the network. In this sense, we can modify the total variance of the network by varying the CV.

We generated random variates from these lognormal distributions for each arc and fitted them to the PH family of distributions (for PA-PH) and a list of conventional travel-time distributions (for PA-MC). For the fitting process of the PH distributions, we used the algorithm proposed by Thummler et al. (2006) with the Java

implementation provided by the jMarkov package (Perez et al. 2017). This algorithm requires as input parameter the number of phases for the fitted PH distribution. In our case, balancing the tradeoff between computational effort and precision, we fit distributions with three and five phases for each arc. For the fitting process of the time-travel distributions we used Lognormal, Gamma, and Weibull distributions, as they are often used in the literature (Zang et al. 2018). Then, we chose the distribution that minimized the χ^2 estimator of the goodness-of-fit test.

Given a network $\mathcal{G} = (\mathcal{N}, \mathcal{A})$ with the cost of each arc c_{ij} and the fitted distribution of travel times, we characterize an instance by its start node v_s , its end node v_e , its time budget T , and its reliability α . To stress the algorithm we must choose a combination of time budget T and reliability α that makes it challenging to find a path that satisfies our chance constraint for a given (v_s, v_e) pair. To design such instances, we take advantage of the fact that we can easily obtain the minimum cost path (\mathcal{P}^c), and the least expected travel time path (\mathcal{P}^t), by solving deterministic shortest path problems. After finding these paths, we use their travel time random variables, $\tilde{t}(\mathcal{P}^c)$ and $\tilde{t}(\mathcal{P}^t)$ to compute their α -quantiles, denoted by $T^c(\alpha)$ and $T^t(\alpha)$, respectively (see Fig. 2). By using these values we can define a tightness factor ($\gamma \in [0, 1]$) proposed by Santos et al. (2007) to set the time budget T for a given (v_s, v_e, α) triplet as follows:

$$T = T^t(\alpha) + (T^c(\alpha) - T^t(\alpha))(1 - \gamma).$$

Finally, PA-PH was implemented in Java and compiled with Eclipse SDK version 4.12. The original PA and PA-MC were implemented in Python version 3.7.6. All experiments were executed on an Intel Xeon E5-2673 v4 @2.30GHz with 56GB of RAM.

5.2 Solving the S- α RP on the Chicago-sketch network

With the procedure described in Sect. 5.1 we derived 20 instances from the Chicago-sketch network (Stabler 2016). This middle-sized network with 933 nodes

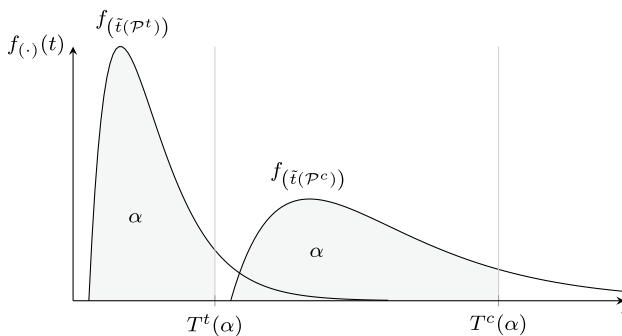


Fig. 2 Time budget selection

Table 1 Chicago-sketch instances

Instance	Source (v_s)	Target (v_e)	Time budget (T)
S- α RP-1	761	376	2307
S- α RP-2	217	268	4411
S- α RP-3	897	477	3826
S- α RP-4	274	84	3412
S- α RP-5	478	448	2171
S- α RP-6	818	70	4638
S- α RP-7	918	159	7152
S- α RP-8	788	488	4073
S- α RP-9	663	902	1779
S- α RP-10	865	757	4139
S- α RP-11	221	195	1724
S- α RP-12	517	444	2850
S- α RP-13	143	417	2445
S- α RP-14	542	750	6468
S- α RP-15	450	223	5125
S- α RP-16	729	353	5852
S- α RP-17	216	61	2973
S- α RP-18	211	144	3172
S- α RP-19	852	831	2668
S- α RP-20	735	382	5003

and 2950 arcs is a fairly realistic, yet aggregate, representation of the Chicago area (Chen and Nie 2015; Li et al. 2010; Nie and Wu 2009). To generate these instances we chose 20 random (v_s, v_e) pairs and the time budget T for a target threshold of $\alpha = 0.90$, $\gamma = 0.4$ and global CV = 0.8. Table 1 lists the (v_s, v_e) pair and the time budget for the 20 instances.

For this experiment, let us recall that we have assumed that travel times follow a lognormal distribution as described in Sect. 5.1. For each arc $(i, j) \in \mathcal{A}$ we have a sample of data generated from these lognormal distributions. Algorithm PA only uses the expected values for the time travel of each arc $(i, j) \in \mathcal{A}$, and disregards the shape of the distributions. In contrast, algorithms PA-MC and PA-PH-3/5 fit the distributions on the observed realizations of $\tilde{t}_{i,j}$'s and make an estimate of the $\mathbb{P}(\tilde{t}(\mathcal{P}) \leq T)$ based on fitted Monte Carlo simulation or the PH distributions, respectively. After the optimal paths are found we evaluate their reliability by sampling the lognormal distributions of the $\tilde{t}_{i,j}$'s (10,000 evaluations). We call this the ex-post evaluation, because we already know the optimal path and the underlying distribution. To assess the quality of the probability estimates, for the PA-MC and PA-PH-3/5, we report the ex-ante estimate, the estimation of the reliability found with the fitted distributions. Note that this ex-ante estimate of the reliability is the estimate used by the algorithms (PA-MC, PA-PH-3/5).

In Table 2, we report in column 1, the instance name; columns 2, 4, 6, and 8 report the ex-post reliability for PA, PA-MC, PA-PH-3, and PA-PH-5, respectively;

Table 2 Ex-ante (fit) vs. ex-post (real) reliability ($\mathbb{P}(\tilde{r}(\mathcal{P}) \leq T)$) for a target threshold of $\alpha = 0.90$

Instance	CSP	S- α RP($\alpha = 0.90$)						Avg. reliability improvement (%)
	PA	PA-MC		PA-PH-3		PA-PH-5		
	Ex-post	Ex-ante	Ex-post	Ex-ante	Ex-post	Ex-ante	Ex-post	
S- α RP-1	0.7785	0.9995	0.9989	0.9977	0.9994	0.9996	0.9991	28.34
S- α RP-2	0.6829	1.0000	0.9999	0.9863	0.9997	0.9974	0.9999	46.41
S- α RP-3	0.6892	1.0000	1.0000	0.9963	1.0000	0.9996	1.0000	45.10
S- α RP-4	0.6224	0.9986	0.9984	0.9284	0.9989	0.9046	0.9837	59.65
S- α RP-5	0.7307	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	36.86
S- α RP-6	0.6517	0.9103	0.9233	0.9744	1.0000	0.9933	1.0000	49.52
S- α RP-7	0.7387	0.9966	0.9956	0.9543	0.9976	0.9832	0.9971	34.94
S- α RP-8	0.7491	0.9128	0.9098	0.9306	0.9878	0.9602	0.9927	28.61
S- α RP-9	0.7803	0.9150	0.7659	0.9862	1.0000	0.9969	1.0000	18.16
S- α RP-10	0.6297	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	58.81
S- α RP-11	0.7410	0.9999	1.0000	0.9998	0.9999	1.0000	0.9999	34.94
S- α RP-12	0.6731	0.9893	0.9825	0.9806	0.9828	0.9924	0.9842	46.07
S- α RP-13	0.7632	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	31.03
S- α RP-14	0.7564	0.9598	0.9639	0.9918	0.9984	0.9727	0.9819	29.75
S- α RP-15	0.7688	1.0000	1.0000	0.9925	1.0000	0.9990	1.0000	30.07
S- α RP-16	0.7754	0.9167	0.9169	0.9907	0.9983	0.9977	0.9975	25.21
S- α RP-17	0.5827	0.9956	0.9956	0.9113	0.9958	0.9402	0.9952	70.85
S- α RP-18	0.8017	0.9999	0.9998	0.9971	0.9998	0.9995	1.0000	24.72
S- α RP-19	0.7400	0.9986	0.9974	0.9987	0.9976	0.9989	0.9983	34.83
S- α RP-20	0.7157	0.9959	0.9955	0.9826	0.9961	0.9936	0.9955	39.12
Avg. Error		- 0.95%		1.77%		0.99%		38.65

columns 3, 5, and 7 report the ex-ante reliability for PA-MC, PA-PH-3, and PA-PH-5, respectively; column 9 shows the average reliability of the paths found for the S- α RP against the reliability of the paths found for the CSP. This reliability improvement is computed as the difference (in %) of the ex-post reliability of the optimal path found for the CSP, and the average ex-post reliability of the paths found for the S- α RP by the three proposed variants. Finally, as a measure of the accuracy of the ex-ante estimation of the reliability, we report on the last row of Table 2 the average percentage error of the ex-ante and ex-post estimation of the reliability for each of the variants.

As expected, from Table 2, we can see that none of the paths found by the PA meets the reliability constraint of the S- α RP. In contrast, the PA-MC finds 90%-reliable solutions in 19 out of the 20 instances. Although PA-MC provides an accuracy within 0.95% of error on average, PA-MC tends to overestimate the reliability in 18 out of 20 instances. In general, PA-MC performs well, yet in instance 9 it estimates a reliability of 91.50% (> 90%) when the ex-post evaluation shows

Table 3 Cost of the α -reliable paths for the Chicago-sketch instances

Instance	CSP	S- α RP ($\alpha = 0.90$)					
	PA	PA-MC	% Δ (vs. CSP) (%)	PA-PH-3	% Δ (vs. CSP) (%)	PA-PH-5	% Δ (vs. CSP) (%)
S- α RP-1	4485	4517	0.71	4517	0.71	4517	0.71
S- α RP-2	8024	8115	1.13	8115	1.13	8115	1.13
S- α RP-3	8231	9072	10.22	9072	10.22	9072	10.22
S- α RP-4	6537	6557	0.31	6557	0.31	6557	0.31
S- α RP-5	4244	4665	9.92	4665	9.92	4665	9.92
S- α RP-6	7992	8218	2.83	8313	4.02	8313	4.02
S- α RP-7	11,037	11,040	0.03	11,040	0.03	11,040	0.03
S- α RP-8	5340	5395	1.03	5417	1.44	5417	1.44
S- α RP-9	3027	3027	0.00	3042	0.5	3042	0.5
S- α RP-10	8548	8687	1.63	8687	1.63	8687	1.63
S- α RP-11	3548	3578	0.85	3578	0.85	3578	0.85
S- α RP-12	6435	6468	0.51	6468	0.51	6468	0.51
S- α RP-13	5630	5779	2.65	5779	2.65	5779	2.65
S- α RP-14	7475	7506	0.41	7593	1.58	7498	0.31
S- α RP-15	9057	9145	0.97	9145	0.97	9145	0.97
S- α RP-16	12,200	12,313	0.93	12,318	0.97	12,318	0.97
S- α RP-17	5124	5133	0.18	5193	1.35	5133	0.18
S- α RP-18	6009	6030	0.35	6030	0.35	6030	0.35
S- α RP-19	5542	5557	0.27	5557	0.27	5557	0.27
S- α RP-20	11,555	12,257	6.08	12,257	6.08	12,257	6.08
Avg. reliability cost			2.05		2.27		2.15

a reliability of just 76.59%, well below 90%. Similarly, PA-PH provides 90%-reliable path for all instances, both for three and five phases. In contrast to PA-MC, PA-PH-3/5 tends to underestimate the reliability, being more conservative overall and avoiding mistakes like the one PA-MC makes in instance 9. In summary, Table 2 shows how the α -reliable paths found for the S- α RP improve the on-time arrival probability by 38.6% (on average) over the optimal paths found by the original PA for the CSP.

Table 3 compares the costs of the optimal paths found for the CSP, with the traditional PA, against the cost of optimal paths found for the S- α RP, with the proposed variants of the PA. Column 1 reports the instance name; column 2 shows the cost of the optimal path for the CSP found with the original pulse algorithm—PA—; columns 3, 5, and 7 report the cost of the optimal path found for the S- α RP with the PA with Monte Carlo simulation—PA-MC—(column 3) and the PA with PH distributions with three and five phases,—PA-PH-3 and PA-PH-5—(columns 5 and 7), respectively. Finally, columns 4, 6, and 8 show the relative difference between the cost of the optimal path found for the CSP and the S- α RP.

Table 4 Computational times for the Chicago-sketch instances (in seconds)

Instance	CSP	S- α RP ($\alpha = 0.90$)					
		PA-MC	PA-PH-3	PA-PH-5	PA-PH-5 vs. PA-PH-3		
S- α RP-1	0.0156	22.6841	0.0262	0.0616	865.0603	368.4414	2.3479
S- α RP-2	0.0156	43.4846	8.5599	62.1376	5.0801	0.6998	7.2592
S- α RP-3	0.0156	151.6370	0.2376	12.1109	638.1545	12.5207	50.9681
S- α RP-4	0.0156	69.9828	97.4334	140.3443	0.7183	0.4987	1.4404
S- α RP-5	0.0156	28.4259	0.0053	0.0249	5385.6295	1141.8588	4.7165
S- α RP-6	0.0156	350.2113	377.5690	1108.7526	0.9275	0.3159	2.9366
S- α RP-7	0.0156	293.3984	38.1748	53.4018	7.6857	5.4942	1.3989
S- α RP-8	0.0156	57.7415	0.0220	0.0745	2623.6123	774.8396	3.3860
S- α RP-9	0.0156	10.4066	0.0120	0.0299	868.3040	348.1664	2.4939
S- α RP-10	0.0156	80.4857	0.0503	0.7819	1600.5615	102.9426	15.5481
S- α RP-11	0.0156	6.6508	0.0023	0.0079	2946.7599	839.3389	3.5108
S- α RP-12	0.0156	7.1096	0.0032	0.0101	2213.4606	703.3394	3.1471
S- α RP-13	0.0000	9.4535	0.0160	0.0717	589.3494	131.8605	4.4695
S- α RP-14	0.0156	74.8461	0.0300	0.2682	2497.3241	279.0778	8.9485
S- α RP-15	0.0156	318.6349	42.5379	67.2473	7.4906	4.7383	1.5809
S- α RP-16	0.0000	204.9908	0.1036	0.4297	1979.0155	477.0851	4.1481
S- α RP-17	0.0156	58.2365	0.6735	35.2501	86.4665	1.6521	52.3376
S- α RP-18	0.0156	17.1486	0.0095	0.7722	1800.9607	22.2069	81.0990
S- α RP-19	0.0000	12.1723	0.0072	0.0182	1682.3271	667.2647	2.5212
S- α RP-20	0.0000	36.7606	0.0381	0.1402	964.8667	262.1579	3.6805
Overall avg.	0.0243	45.4965	0.1613	0.8623	282.0389	52.7589	5.3458
Total fitting time	-	780.3314	1765.9260	5665.7541			

Since the CSP is a relaxation of the S- α RP, the optimal cost is always less than or equal to the optimal cost for the S- α RP. The difference between the cost of the CSP and the S- α RP can be interpreted as the cost we are willing to pay for a reliable path. One of the advantages of using the α -reliable paths found by solving the S- α RP, is that just by paying an extra 2.1% one can find alternative paths with large on-time arrival probabilities.

Finally, Table 4 compares the proposed variants for solving the S- α RP in terms of computational times. Column 1 shows the instance name; columns 2–5 report the computational time in seconds achieved with PA, PA-MC, and PA-PH-3/5, respectively. In addition, columns 6 and 7 show the speedups achieved by PA-PH-3/5 against PA-MC, respectively; and column 8 show the speedups achieved by PA-PH-3 against PA-PH-5. The overall averages represent the geometric means of the respective column. Finally, in the last row we present the computational time required to fit the distributions for each of the variants of the PA.

Because PA-MC and PA-PH used different languages (Python and Java, respectively), we avoid a direct comparison. Overall, as expected, the deterministic PA runs much faster than the PA-MC and PA-PH-3/5. For instance, PA runs on average 564% faster than PA-PH-3, the best performer among the two PH implementations (in terms of computational time). Note that adding more phases improves the reliability accuracy by 44.07% (dropping the average error from 1.77 to 0.99%, as seen in Table 2), yet this precision comes at a fivefold increment in computational cost (Table 4). Here, it is also important to consider the fitting times for each of the PA-PH variants. In the case we use the PH distributions, improving the accuracy does not only imply a greater computational cost while doing the search, but also the preprocessing time takes twice as longer when fitting five rather than three phases. In contrast, the preprocessing times for PA-MC is 1.2 times faster than PA-PH-3, the best performer among the two PH implementations (in terms of preprocessing time).

5.3 Illustrative example of spatially-correlated travel times

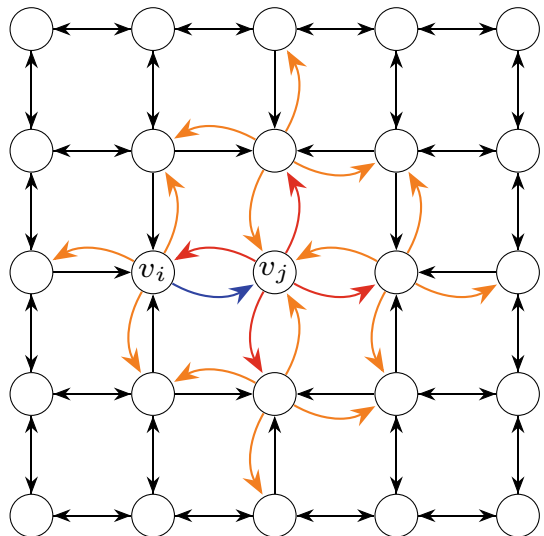
In real transportation networks, travel times are likely to be correlated (Zeng et al. 2015; Huang and Gao 2012; Zhang 2019; Bobbio et al. 2005; Xing and Zhou 2011). To illustrate the impact of including spatial correlations, we implemented PA-MC with the procedure described in Sect. 4.2. With the procedure described in Sect. 5.1 we derived 20 instances from the Sioux Falls road network (24 nodes and 76 arcs) (Stabler 2016) by choosing 20 random (v_s, v_e) pairs and the time budget T for a target threshold of $\alpha = 0.95$, $\gamma = 0.5$ and $CV = 1.2$. Table 5 lists the (v_s, v_e) pair and the time budget for the 20 instances.

To account for the correlation structure, we defined a correlation matrix Σ_G for the arc travel times of the network. Since we are considering spatial correlations for each arc $a = (v_i, v_j) \in \mathcal{A}$, we define sets of arcs at two adjacency levels, $\mathcal{L}_1(a)$ and $\mathcal{L}_2(a)$; these sets are defined as follows: (1) $\mathcal{L}_1(a) = \left\{ a' = (v'_i, v'_j) \in \mathcal{A} : v'_i = v_j \right\}$ and (2) $\mathcal{L}_2(a) = \bigcup_{a' \in \mathcal{L}_1(a)} \mathcal{L}_1(a')$. Figure 3 shows for an arc $a = (v_i, v_j)$ (in blue), the arcs at $\mathcal{L}_1(a)$ (in red) and the arcs at

Table 5 Sioux falls instances

Instance	Source (v_s)	Target (v_e)	Time budget (T)
S- α RP- ρ -1	13	7	3426
S- α RP- ρ -2	12	7	2819
S- α RP- ρ -3	18	12	3352
S- α RP- ρ -4	15	24	1948
S- α RP- ρ -5	4	19	3467
S- α RP- ρ -6	14	8	2949
S- α RP- ρ -7	13	16	3676
S- α RP- ρ -8	3	6	1149
S- α RP- ρ -9	3	20	3354
S- α RP- ρ -10	8	15	2004
S- α RP- ρ -11	7	5	2354
S- α RP- ρ -12	6	19	3110
S- α RP- ρ -13	19	4	3482
S- α RP- ρ -14	19	1	3434
S- α RP- ρ -15	14	3	2184
S- α RP- ρ -16	4	20	3020
S- α RP- ρ -17	10	2	3879
S- α RP- ρ -18	10	20	2132
S- α RP- ρ -19	4	15	2405
S- α RP- ρ -20	8	12	2755

Fig. 3 Arcs correlated with $a = (v_i, v_j) \in \mathcal{A}$ (in blue): set of arcs $\mathcal{L}_1(a)$ in red; set of arcs $\mathcal{L}_2(a)$ in orange (color figure online)



$\mathcal{L}_2(a)$ (in orange). For an arc $a = (v_i, v_j) \in \mathcal{A}$, we induce the correlation coefficients ρ and $\frac{\rho}{2}$ between arcs at the first and second adjacency levels, respectively. Based on these correlations, the covariance matrix Σ_G is defined as follows:

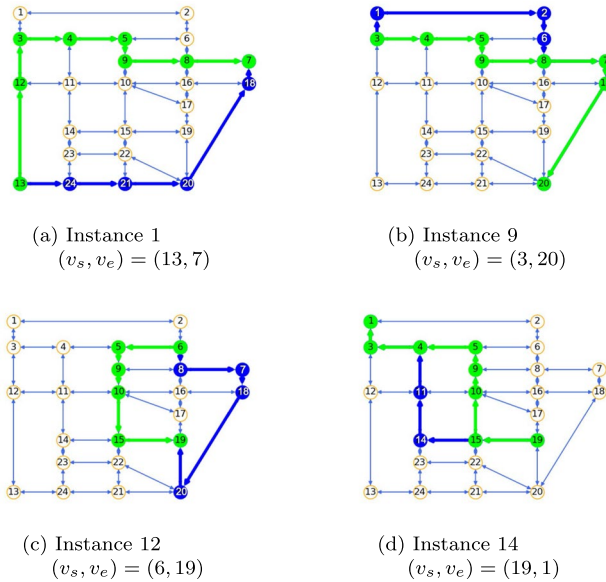


Fig. 4 Reliable paths obtained with PA-MC with and without correlation. Green path: with correlation ($\rho = 0.5$); blue path: without correlation (color figure online)

$$\Sigma_{\mathcal{G}_{a,a'}} = \begin{cases} \rho \cdot \sigma(\tilde{t}_a) \cdot \sigma(\tilde{t}_{a'}) & a' \in \mathcal{L}_1(a) \\ \frac{\rho}{2} \cdot \sigma(\tilde{t}_a) \cdot \sigma(\tilde{t}_{a'}) & a' \in \mathcal{L}_2(a) \\ 0 & \text{otherwise} \end{cases}$$

where $\sigma(\cdot)$ denotes the standard deviation of a random variable. Note that we have induced this correlation structure $\Sigma_{\mathcal{G}}$, but in a real context, the matrix $\Sigma_{\mathcal{G}}$ would be the covariance matrix estimated from the sample data of travel times. Given the covariance matrix $\Sigma_{\mathcal{G}}$ and applying the procedure presented by Cario and Nelson (1997), we computed the covariance matrix $\Sigma_{\mathcal{Z}}$ and followed the procedure presented in Sect. 4.2. Figure 4 illustrates the different optimal paths found with and without correlations with PA-MC. Even in this small-sized network, the reliable paths may vary drastically under spatially correlated travel times.

Table 6 shows for each of the instances (column 1) the cost of the optimal path for the S- α RP without and with spatial correlation structure $\Sigma_{\mathcal{G}}$, namely, PA-MC-0 (zero correlation) in column 2 and PA-MC- ρ in column 3, respectively. Additionally column 4 represents the price paid for finding paths with a correlation structure against finding paths ignoring correlations. As we can see in Table 6, in 25% of the instances the optimal path changed between the correlated and uncorrelated scenarios. Note that the paths found by the PA-MC- ρ are 9.37% more expensive on average. This means that under spatially correlated travel times the solution space tends to be tighter and the optimal paths more expensive.

Table 7 reports the *ex-post* estimates of the on-time arrival probabilities of paths found by PA-MC-0 and PA-MC- ρ , respectively. For each instance (column

Table 6 Cost of the optimal path for the Sioux Falls instances ($\rho = 0.5$)

Instance	PA-MC-0	PA-MC- ρ	% Δ (PA-MC-0 vs. PA-MC- ρ) (%)
S- α RP- ρ -1	3211	5239	63.16
S- α RP- ρ -2	4732	4732	0.00
S- α RP- ρ -3	5070	5070	0.00
S- α RP- ρ -4	1859	1859	0.00
S- α RP- ρ -5	3042	3042	0.00
S- α RP- ρ -6	3549	3549	0.00
S- α RP- ρ -7	3380	5577	65.00
S- α RP- ρ -8	2535	2535	0.00
S- α RP- ρ -9	4394	5070	15.38
S- α RP- ρ -10	2704	2704	0.00
S- α RP- ρ -11	3042	3042	0.00
S- α RP- ρ -12	2535	3549	40.00
S- α RP- ρ -13	3042	3042	0.00
S- α RP- ρ -14	4394	4563	3.85
S- α RP- ρ -15	2366	2366	0.00
S- α RP- ρ -16	4394	4394	0.00
S- α RP- ρ -17	2873	2873	0.00
S- α RP- ρ -18	2197	2197	0.00
S- α RP- ρ -19	2704	2704	0.00
S- α RP- ρ -20	4225	4225	0.00
Avg. overcost	304.2		9.37

1), we report in columns 2 and 5 the ex-post reliability considering the correlation structure of the optimal paths found by PA-MC-0 and PA-MC- ρ , respectively; columns 3 and 6 show the ex-post reliability under the assumption of independence found by PA-MC-0 and PA-MC- ρ , respectively; and finally, columns 4 and 7 report the difference between reliabilities with and without correlations. Table 7 highlights the importance of considering correlations when solving the S- α RP. As one can see, the optimal paths found by PA-MC- ρ meet the reliability constraint in both cases: a scenario in which the travel times are spatially correlated, and a scenario in which this random travel times are independent. In contrast, PA-MC-0 fails sometimes to deliver reliable routes under the scenario with a spatially correlated network. Although in several instances the ex-post reliability meets the target threshold ($\alpha = 0.95$ in this case), in some instances these paths fail to satisfy the reliability constraint under correlated travel times (see instances 1, 7, 9, 12, and 14). Nevertheless, the robustness of the reliable paths found by PA-MC- ρ comes at an average overcost of 9.37% (Table 6).

Finally, Table 8 presents the computational times. Column 1 shows the instance name; columns 2 and 3 report the computational time in seconds for PA-MC-0 and PA-MC- ρ , respectively; and column 4 shows the speedup achieved by PA-MC- ρ over

Table 7 Ex-post (ρ) vs. ex-post reliability ($\mathbb{P}(\tilde{t}(\mathcal{P}) \leq T)$) for a target threshold of $\alpha = 0.95$

Instance	PA-MC-0			PA-MC- ρ		
	Ex-post (ρ)	Ex-post (0)	% Δ (0 vs. ρ) (%)	Ex-post (ρ)	Ex-post (0)	% Δ (0 vs. ρ) (%)
S- α RP- ρ -1	0.8670	0.9508	9.67	0.9998	0.9999	0.01
S- α RP- ρ -2	0.9866	0.9870	0.04	0.9872	0.9894	0.22
S- α RP- ρ -3	0.9959	1.0000	0.41	0.9966	0.9999	0.33
S- α RP- ρ -4	0.9820	0.9638	1.86	0.9836	0.9655	1.84
S- α RP- ρ -5	0.9763	0.9987	2.30	0.9784	0.9982	2.02
S- α RP- ρ -6	0.9698	0.9935	2.45	0.9712	0.9933	2.28
S- α RP- ρ -7	0.9008	0.9743	8.16	0.9970	0.9993	0.23
S- α RP- ρ -8	0.9973	1.0000	0.27	0.9978	1.0000	0.22
S- α RP- ρ -9	0.8866	0.9539	7.60	0.9998	1.0000	0.02
S- α RP- ρ -10	0.9845	0.9962	1.19	0.9840	0.9957	1.19
S- α RP- ρ -11	0.9831	0.9994	1.66	0.9850	0.9992	1.44
S- α RP- ρ -12	0.9436	0.9731	3.12	0.9568	0.9669	1.06
S- α RP- ρ -13	0.9622	0.9979	3.71	0.9620	0.9981	3.75
S- α RP- ρ -14	0.8717	0.9671	10.95	0.9870	0.9995	1.27
S- α RP- ρ -15	0.9644	0.9819	1.82	0.9662	0.9809	1.52
S- α RP- ρ -16	0.9998	0.9999	0.01	0.9996	1.0000	0.04
S- α RP- ρ -17	0.9996	1.0000	0.04	0.9996	1.0000	0.04
S- α RP- ρ -18	0.9839	0.9749	0.91	0.9844	0.9783	0.62
S- α RP- ρ -19	0.9705	0.9840	1.39	0.9714	0.9831	1.20
S- α RP- ρ -20	0.9939	0.9999	0.61	0.9952	1.0000	0.48
Overall Avg.			2.91			0.99

PA-MC-0. The overall averages represent the geometric mean of the computational times in its respective column. Surprisingly, the computational times drop drastically when choosing PA-PH- ρ over PA-MC-0 to solve the S- α RP. The computational times when solving the S- α RP with correlation structure is on average seven times faster than under the assumption of independence. This may be due to the fact that the correlated random travel times help the algorithm prune pulses early and leads to explore less paths. In terms of preprocessing, note that fitting times are the same in both variants, since they share the marginal distributions. The difference arises in the estimation of the covariance matrix $\Sigma_{\mathcal{Z}}$ for the NORTA procedure. This covariance matrix estimation relies on a numerical computation of double integrals that can be expensive. Note that this covariance matrix has dimensions $|\mathcal{A}| \times |\mathcal{A}|$, where \mathcal{A} is the set of arcs of the underlying network, so obtaining this target covariance matrix $\Sigma_{\mathcal{Z}}$ for the NORTA procedure may pose some computational challenges.

5.4 Scenario-based experiments

In transportation networks the travel time distributions are likely to depend on bad weather, rush hours, or particular traffic patterns, among other factors. In this

Table 8 Computational times for the Sioux Falls instances (in seconds)

Instance	PA-MC-0	PA-MC- ρ	Speedup
S- α RP- ρ -1	6.2660	5.5003	1.1392
S- α RP- ρ -2	28.7989	4.1095	7.0079
S- α RP- ρ -3	81.5053	8.0790	10.0885
S- α RP- ρ -4	7.7974	0.7344	10.6179
S- α RP- ρ -5	18.9856	2.4376	7.7888
S- α RP- ρ -6	20.6106	2.4689	8.3482
S- α RP- ρ -7	81.9118	9.4378	8.6791
S- α RP- ρ -8	4.1407	0.5939	6.9716
S- α RP- ρ -9	59.7852	6.4067	9.3317
S- α RP- ρ -10	19.8450	2.2034	9.0065
S- α RP- ρ -11	7.9068	1.0004	7.9037
S- α RP- ρ -12	15.9698	2.7817	5.7410
S- α RP- ρ -13	15.5010	2.2975	6.7470
S- α RP- ρ -14	51.4252	7.0629	7.2810
S- α RP- ρ -15	2.9374	0.4062	7.2313
S- α RP- ρ -16	29.7208	3.7191	7.9915
S- α RP- ρ -17	17.9077	3.5784	5.0044
S- α RP- ρ -18	5.2815	0.5938	8.8940
S- α RP- ρ -19	10.2038	1.3126	7.7736
S- α RP- ρ -20	19.1423	2.3591	8.1142
Overall Avg.	16.7245	2.3598	7.0874

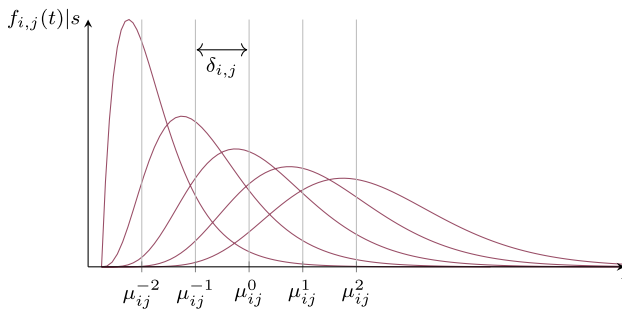


Fig. 5 Conditioned distributions for arc $(i, j) \in \mathcal{A}$ with $\mathcal{S} = \{-2, -1, 0, 1, 2\}$ and mean μ_{ij}^s

section, we illustrate how our proposed algorithm obtains costs and reliability estimates using scenario-dependent arc travel times.

More specifically, we condition the travel times on a particular scenario s in a set of scenarios \mathcal{S} . We define a discrete probability distribution over the set of scenarios, where p_s represents the probability of the whole network being under scenario $s \in \mathcal{S}$. Given a scenario, we assume that the arc travel times are independent. For each arc $(i, j) \in \mathcal{A}$ we draw travel times from its scenario distribution. In other words, by changing the distribution parameters of each arc based on a given scenario, we simulate

different network conditions such as heavy vs. light traffic. Figure 5 shows an example of different distributions for a specific arc, under five different scenarios.

For the experiments in this section, we adapted the Chicago-sketch road network (933 nodes and 2950 arcs) from Stabler (2016). We assume that all travel times, in every scenario, follow a lognormal distribution with mean $\mu_{(i,j)}^s$ and standard deviation $\sigma_{(i,j)}^s$, which need to be determined. For each arc, we assume its unconditional expected travel time to be the value reported in the repository (i.e., $\mathbb{E}[\tilde{t}_{(i,j)}]$). The means $\mu_{(i,j)}^s$ are fixed, such that

$$\sum_{s \in \mathcal{S}} p_s \cdot \mu_{(i,j)}^s = \mathbb{E}[\tilde{t}_{(i,j)}] \tag{5.1}$$

holds, where $\mu_{(i,j)}^s \triangleq \mathbb{E}[\tilde{t}_{(i,j)}|s]$ for $s \in \mathcal{S}$. To choose the values of $\mu_{(i,j)}^s$ we set an integer $n \in \mathbb{N}$ and take $\mathcal{S} = \{s \in \mathbb{Z} : |s| \leq n\}$, where scenario $s = 0$ represents the neutral scenario, with $\mu_{(i,j)}^0 = \mathbb{E}[\tilde{t}_{(i,j)}]$ for all arc $(i,j) \in \mathcal{A}$. We assume a symmetric discrete distribution for p_s , e.g., $p_s = \frac{1}{|S|}$, and define a step size $\delta_{i,j} = \frac{\mathbb{E}[\tilde{t}_{(i,j)}] - t_{ij}}{n+1}$. Then the conditional means that make Eq. (5.1) hold can be expressed as $\mu_{(i,j)}^s = \mathbb{E}[\tilde{t}_{(i,j)}] + s \cdot \delta_{i,j}$. For each arc we assume a fixed coefficient of variation (CV) of the shifted distribution. Therefore, for each arc $(i,j) \in \mathcal{A}$ and scenario $s \in \mathcal{S}$, the standard deviation is assumed to be $\sigma_{(i,j)}^s = CV \cdot \mu_{(i,j)}^s$.

Table 9 Scenario approach instances

Instance	Source (v_s)	Target (v_e)	Time budget (T)
S- α RP-S-1	393	897	8104
S- α RP-S-2	660	920	7706
S- α RP-S-3	849	378	9347
S- α RP-S-4	526	781	8318
S- α RP-S-5	561	239	6247
S- α RP-S-6	723	911	7502
S- α RP-S-7	893	379	9344
S- α RP-S-8	171	892	6469
S- α RP-S-9	792	898	10,587
S- α RP-S-10	213	910	8399
S- α RP-S-11	45	899	7152
S- α RP-S-12	886	255	8634
S- α RP-S-13	736	418	5794
S- α RP-S-14	241	383	10,435
S- α RP-S-15	850	374	8263
S- α RP-S-16	258	365	9191
S- α RP-S-17	727	362	6877
S- α RP-S-18	531	239	7210
S- α RP-S-19	823	198	6292
S- α RP-S-20	415	442	5762

With the procedure described in Sect. 5.1 we derived 20 instances from the Chicago-sketch road network (933 nodes and 2950 arcs) (Stabler 2016) by choosing 20 random (v_s, v_e) pairs and the time budget T for a target threshold of $\alpha = 0.80$, $\gamma = 0.2$ and a CV = 0.8. Table 9 lists the (v_s, v_e) pair and the time budget for the 20 instances. The experiments in this section aim to estimate the expected cost $\bar{c}(v_s, v_e)$ and the expected reliability $\bar{\pi}(v_s, v_e)$ over all scenarios when going from v_s to v_e . To this end, we solve for each instance and each scenario the S- α RP, and then estimate $\bar{c}(v_s, v_e)$ and $\bar{\pi}(v_s, v_e)$ as

$$\bar{c}(v_s, v_e) = \sum_{s \in \mathcal{S}} p_s c \cdot (\mathcal{P}_{v_s, v_e}^s), \quad \bar{\pi}(v_s, v_e) = \sum_{s \in \mathcal{S}} p_s \cdot \pi(\mathcal{P}_{v_s, v_e}^s), \quad (5.2)$$

where \mathcal{P}_{v_s, v_e}^s is the optimal path found when solving the S- α RP for scenario $s \in \mathcal{S}$. For the experiments in this section we used PA-PH3 on five scenarios ($n = 2$). For the sake of clarity, henceforth the scenario $s \in \mathcal{S}$ will be denoted by $s \cdot \delta$ (e.g., scenario -2 will be denoted by -2δ).

In Table 10, we report in column 1, the instance name; columns 2–6 report the costs of the paths found by PA-PH3 for the respective scenario; and column 7 reports the estimate of $\bar{c}(v_s, v_e)$ based on the total probability theorem (in

Table 10 Cost for the scenario approach and a target threshold of $\alpha = 0.80$

Instance	Scenario					$\bar{c}(v_s, v_e)$
	-2δ	$-\delta$	0	δ	2δ	
S- α RP-S-1	13,517	13,517	13,517	13,547	13,735	13,567
S- α RP-S-2	12,863	12,863	12,863	12,885	12,887	12,872
S- α RP-S-3	16,375	16,375	16,375	16,412	16,629	16,433
S- α RP-S-4	14,620	14,620	14,620	14,627	15,946	14,887
S- α RP-S-5	11,595	11,595	11,595	11,602	11,945	11,666
S- α RP-S-6	13,823	13,823	13,823	13,996	14,982	14,089
S- α RP-S-7	19,118	19,118	19,118	19,148	19,365	19,173
S- α RP-S-8	13,169	13,169	13,169	13,174	13,306	13,197
S- α RP-S-9	16,761	16,761	16,761	16,798	16,966	16,809
S- α RP-S-10	14,702	14,702	14,702	14,732	15,044	14,776
S- α RP-S-11	12,781	12,781	12,872	12,872	13,060	12,873
S- α RP-S-12	16,094	16,094	16,094	16,131	16,214	16,125
S- α RP-S-13	11,131	11,131	11,131	11,172	11,332	11,179
S- α RP-S-14	23,744	23,744	24,066	24,137	25,201	24,178
S- α RP-S-15	14,694	14,694	14,694	14,824	15,334	14,848
S- α RP-S-16	15,450	15,450	15,450	15,487	15,799	15,527
S- α RP-S-17	12,600	12,600	12,600	12,641	12,843	12,657
S- α RP-S-18	12,944	12,944	12,944	12,951	13,294	13,015
S- α RP-S-19	11,538	11,538	11,538	11,579	11,693	11,577
S- α RP-S-20	11,755	11,755	11,760	11,760	11,878	11,782
Avg. % Δ vs. 0	- 0.10%	- 0.10%	-	0.27%	2.60%	0.53%

Table 11 Reliability for the scenario approach and a target threshold of $\alpha = 0.80$

Instance	Scenario					$\bar{\pi}(v_s, v_e)$
	-2δ	$-\delta$	0	δ	2δ	
S- α RP-S-1	1.0000	0.9973	0.8512	0.8341	0.8904	0.9146
S- α RP-S-2	1.0000	0.9989	0.8042	0.9541	0.8609	0.9236
S- α RP-S-3	1.0000	1.0000	0.9530	0.9682	0.8345	0.9511
S- α RP-S-4	1.0000	1.0000	0.9879	0.8398	0.8327	0.9321
S- α RP-S-5	1.0000	0.9942	0.8055	0.8594	0.9544	0.9227
S- α RP-S-6	1.0000	1.0000	0.9971	0.8420	0.9355	0.9549
S- α RP-S-7	1.0000	1.0000	0.9907	0.9659	0.8073	0.9528
S- α RP-S-8	1.0000	1.0000	0.8981	0.8893	0.8166	0.9208
S- α RP-S-9	1.0000	0.9984	0.8457	0.8311	0.8767	0.9104
S- α RP-S-10	1.0000	1.0000	0.9854	0.9873	0.8011	0.9548
S- α RP-S-11	1.0000	0.9769	0.9691	0.8303	0.9858	0.9524
S- α RP-S-12	1.0000	0.9998	0.8470	0.9179	0.9809	0.9491
S- α RP-S-13	1.0000	1.0000	0.9952	0.8300	0.8012	0.9253
S- α RP-S-14	1.0000	0.9652	0.8113	0.9687	0.8163	0.9123
S- α RP-S-15	1.0000	1.0000	0.9953	0.8285	0.9237	0.9495
S- α RP-S-16	1.0000	1.0000	0.9753	0.9804	0.8038	0.9519
S- α RP-S-17	1.0000	1.0000	0.9712	0.8214	0.8310	0.9247
S- α RP-S-18	1.0000	1.0000	0.9853	0.8661	0.8378	0.9378
S- α RP-S-19	1.0000	1.0000	0.9834	0.8003	0.9697	0.9507
S- α RP-S-20	1.0000	0.9981	0.9957	0.8758	0.8447	0.9429

(5.2)). Finally, the last row reports the average cost difference against the neutral scenario.

In Table 11, we report in column 1, the instance name; columns 2–6 report the ex-ante reliability of the paths found by the proposed algorithm for the respective scenario; and column 7 reports the estimation of $\bar{\pi}(v_s, v_e)$ through the total probability theorem (in (5.2)).

As observed in Table 10, the cost of the paths found by the algorithm increases as the scenario is stressed (by the increment on the expected value of the arc travel times). Note that the more stressed the scenario is, the fewer reliable paths exist; in other words, the solution space reduces and the objective function deteriorates. However, the same monotonicity does not hold when assessing the reliability. As seen in Table 11, in many instances the path found for scenario 2δ is more reliable than the path found for the neutral scenario (see instances S- α RP-S-1/2/5/9/11/12/14). Therefore, the reliability of the more stressed scenarios comes at a cost. As shown in Table 10 the paths found for scenario 2δ have an average of 2.60% overcost relative to the neutral scenario.

6 Conclusions

In this work we extended the pulse algorithmic framework to solve the shortest α -reliable path problem. The S - α RP is modeled as a chance-constrained shortest path problem with random arc travel times. To treat uncertainty, we propose two approaches: the first one models travel times as independent Phase-type distributed random variables (PA-PH); and the second approach uses Monte-Carlo simulation to model travel times with arbitrary distributions with a correlation structure (PA-MC).

We conducted a series of experiments over road networks widely used in the transportation literature. The α -reliable paths found for the S - α RP with PA-MC and PA-PH improve the on-time arrival probability by 38.6% (on average) over the optimal paths found by a pulse algorithm that uses expected travel times and ignores the inherent randomness. This dramatic improvement in reliability comes at the expense of a 2.1% marginal increment in cost. We observed that adding more phases (from three to five) in the PA-PH improves the reliability accuracy, dropping the average error from 1.77 to 0.99%. However, this precision comes at a hefty price of a fivefold increment in computational cost. Having more phases also takes a toll in preprocessing, as PH distributions with five phases takes twice as longer to fit than those with three phases. In contrast, PA-MC takes 1.2 less preprocessing than the best performer PA-PH-3. When considering correlated travel times, the reliable paths found by PA-MC- ρ came at an average overcost of 9.37%.

We also introduced an alternative scenario-based approach where it is possible to model some situations where correlated travel times naturally arise, such as those found in urban transportation settings where external factors such as the weather and traffic influencing the whole state of the network. This approach provides an alternative to tackle correlated arc travel times by grouping them into scenarios and then, applying the total probability theorem to compute cost and reliability estimates (across scenarios). It is worth noting that this approach demands more data, as multiple scenarios are built based on the factors affecting the whole network (e.g., time of the day, weather conditions).

The PA-PH and PA-MC developed for the S - α RP open several avenues for future research. For example, we noted that the PA-PH is conservative and in general it underestimates the on-time arrival probabilities. This underestimation causes the algorithm to declare some instances infeasible, when some paths that meet the reliability constraint exist. We also see the opportunity that extended chance-constrained models may be able to handle other risk measures (e.g., VaR or CVaR) in the stochastic constraints. These other variants could be relatively easily handled by the pulse algorithmic framework based on the foundation laid out for the S - α RP. Another line of research could handle variants with stochastic objectives such as finding the most reliable path (Chen and Ji 2005). We leave these considerations for further research.

Appendix

See Table 12.

Table 12 Notation table

Pulse algorithm for the S α RP:

$\mathcal{G} = (\mathcal{N}, \mathcal{A})$	Directed graph
\mathcal{N}	Set of nodes $\mathcal{N} = \{v_1, \dots, v_n\}$
\mathcal{A}	Set of arcs $\mathcal{A} \subseteq \{(i, j) v_i \in \mathcal{N}, v_j \in \mathcal{N}, i \neq j\}$
$\Gamma^+(v_i)$	Forward star of v_i , $\Gamma^+(v_i) = \{v_j \in \mathcal{N} (i, j) \in \mathcal{A}\}$
\mathcal{P}_{ij}	A path from node v_i to v_j . Sequence of nodes $\{v_i, \dots, v_{(k)}, \dots, v_j\}$ or sequence of arcs $\mathcal{P} = \{\dots, (i, j)_{(k)}, \dots\}$
$\mathcal{P}_{i,e}^t$	Minimum-time path from node $v_i \in \mathcal{N}$ to the end node $v_e \in \mathcal{N}$, under the best possible scenario
$\mathcal{P}_{i,e}^c$	Minimum-cost path from node v_i to the end node v_e
Ω_{ij}	Set of all possible paths from node $v_i \in \mathcal{N}$ to node $v_j \in \mathcal{N}$
c_{ij}	Cost of traversing arc $(i, j) \in \mathcal{A}$
$c(\mathcal{P}) \triangleq \sum_{(i,j) \in \mathcal{P}} c_{ij}$	Cost of traversing path \mathcal{P}
\bar{c}	Global primal bound of the minimum cost
$c(i)$	Lower bound on the cost from node $v_i \in \mathcal{N}$ to the end node $v_e \in \mathcal{N}$. Corresponds to the cost of $\mathcal{P}_{i,e}^c$
α	On-time arrival probability (reliability) threshold
T	Maximum time budget
\tilde{t}_{ij}	Random variable of the traversing time of arc $(i, j) \in \mathcal{A}$
t_{ij}	Deterministic part of the traversing time of arc $(i, j) \in \mathcal{A}$ accounting for this free-flow speed time
\hat{t}_{ij}	Stochastic component of the traversing time of arc $(i, j) \in \mathcal{A}$
$t(i)$	Minimum time consumption from any node v_i to the final node v_e under the best possible scenario
$\tilde{t}(\mathcal{P}) \triangleq \sum_{(i,j) \in \mathcal{P}} \tilde{t}_{ij}$	Random variable of the traversing time of path \mathcal{P}
$T(i) \triangleq T - t(i)$	Maximum allowed time consumption for a partial path up to node v_i
$\pi(\mathcal{P}) \triangleq \mathbb{P}(\tilde{t}(\mathcal{P}_{s,i}) \leq T(i)) \leq \alpha$	The probability of arriving on time under the best possible scenario denoted
$L(v_i)$	Limited set of labels to check if an incoming pulse to node $v_i \in \mathcal{N}$ is dominated
R	Total number of labels (i.e., memory size)
Phase-type distributions	
$(\tau_0, \boldsymbol{\tau})$	Initial probability vector
\boldsymbol{Q}	Infinitesimal generator
\boldsymbol{Z}	Infinitesimal generator of the transient states
\boldsymbol{z}	Absorption rates
Monte Carlo simulation	
$\boldsymbol{t}_{\mathcal{P}}$	Random vector, where $\tilde{t}_{(i,j)_{(k)}}$ represents the random travel time of the k -th arc of the path \mathcal{P}
$\mathbf{1}\{\mathbf{1}^T \boldsymbol{t}_{\mathcal{P}} \leq \tilde{t}(v)\}$	Indicator function of the event $\{\mathbf{1}^T \boldsymbol{t}_{\mathcal{P}} \leq \tilde{t}(v)\}$ for $v \in \mathcal{N}$
$\hat{\pi}(\mathcal{P})$	Monte Carlo estimator of the $\pi(\mathcal{P})$
F_k	Cumulative distribution function of $\tilde{t}_{(i,j)_{(k)}}$, the random time of the k -th arc of the path \mathcal{P}
$\text{Cov}[\boldsymbol{t}_{\mathcal{P}}]$	Covariance matrix of $\boldsymbol{t}_{\mathcal{P}}$
\boldsymbol{Z}	$ \mathcal{P} $ -dimensional, standard multivariate normal vector

Table 12 (continued)

Σ_Z	NORTA covariance matrix
Correlated experiments	
Σ_G	Covariance matrix of the arcs in the graph G
$\mathcal{L}_1(a) \triangleq \left\{ a' = (v'_i, v'_j) \in \mathcal{A} : v'_i = v_j \right\}$	First level of adjacency for arc $a = (v_i, v_j) \in \mathcal{A}$
$\mathcal{L}_2(a) \triangleq \bigcup_{a' \in \mathcal{L}_1(a)} \mathcal{L}_1(a')$	Second level of adjacency for arc $a \in \mathcal{A}$

Acknowledgements First and foremost, we would like to thank the editors-in-chief Antonio Alonso-Ayuso and Dolores Romero-Morales and the guest editors Laureano F. Escudero and Nelson Maculan for the invitation to submit to this very special issue dedicated to the Ibero-American operational research community. Special thanks go to Hector Cancela who invited A. Medaglia to teach a course on the pulse framework at Universidad de la República (Uruguay). The active exchange of ideas with students were enlightening and certainly improved parts of this paper. The authors would like to thank the anonymous reviewer who provided insightful ideas to induce correlations via the scenario-based approach. Finally, we would like to thank the partial funding provided by the Office of Research at Universidad de los Andes.

References

- Ahuja RK, Magnanti TL, Orlin JB (1993) Network flows: theory, algorithms, and applications. Prentice Hall, Englewood Cliffs
- Asmussen S, Nerman O, Olsson M (1996) Fitting Phase-Type Distributions via the EM Algorithm. *Scand J Stat* 23(4):419–441
- Bobbio A, Horváth A, Telek M (2005) Matching three moments with minimal acyclic phase type distributions. *Stoch Models* 21(2–3):303–326. <https://doi.org/10.1081/stm-200056210>
- Bolívar MA, Lozano L, Medaglia AL (2014) Acceleration strategies for the weight constrained shortest path problem with replenishment. *Optim Lett* 8(8):2155–2172. <https://doi.org/10.1007/s11590-014-0742-x>
- Cabrera N, Medaglia AL, Lozano L, Duque D (2020) An exact bidirectional pulse algorithm for the constrained shortest path. *Networks* 76(2):128–146
- Cario MC, Nelson BL (1997) Modeling and generating random vectors with arbitrary marginal distributions and correlation matrix. Technical report
- Chen A, Ji Z (2005) Path finding under uncertainty. *J Adv Transp* 39(1):19–37. <https://doi.org/10.1002/atr.5670390104>
- Chen BY, Lam WHK, Sumalee A, Li ZI YuB, Lam WHK, Sumalee A, ZIL Reliable (2012) Reliable shortest path finding in stochastic networks with spatial correlated link travel times. *Int J Geogr Inf Sci* 26(2):365–386. <https://doi.org/10.1080/13658816.2011.598133>
- Chen PW, Nie YM (2015) Stochastic optimal path problem with relays. *Transp Res Procedia* 7:129–148. <https://doi.org/10.1016/j.trpro.2015.06.008>
- Dijkstra E (1959) A note on two problems in connexion with graphs. *Numer Math* 1:269–271. <https://doi.org/10.1007/BF01386390>
- Dumitrescu I, Boland N (2003) Improved preprocessing, labeling and scaling algorithms for the weight-constrained shortest path problem. *Networks* 42(3):135–153. <https://doi.org/10.1002/net.10090>
- Duque D, Medaglia AL (2019) An exact method for a class of robust shortest path problems with scenarios. *Networks* 74(4):360–373. <https://doi.org/10.1002/net.21909>
- Duque D, Lozano L, Medaglia AL (2014) Solving the orienteering problem with time windows via the pulse framework. *Comput Oper Res* 54:168–176. <https://doi.org/10.1016/j.cor.2014.08.019>
- Duque D, Lozano L, Medaglia AL (2015) An exact method for the biobjective shortest path problem for large-scale road networks. *Eur J Oper Res* 242(3):788–797. <https://doi.org/10.1016/j.ejor.2014.11.003>

- Frank H (1969) Shortest paths in probabilistic graphs. *Oper Res* 17(4):583–599. <https://doi.org/10.1287/opro.17.4.583>
- Gallo G, Pallottino S (1988) Shortest path algorithms. *Ann Oper Res* 13(1):1–79. <https://doi.org/10.1007/BF02288320>
- van der Geest P (1998) An algorithm to generate samples of multi-variate distributions with correlated marginals. *Comput Stat Data Anal* 27(3):271–289. [https://doi.org/10.1016/S0167-9473\(98\)00005-X](https://doi.org/10.1016/S0167-9473(98)00005-X)
- Gómez A, Mariño R, Akhavan-Tabatabaei R, Medaglia AL, Mendoza JE (2016) On modeling stochastic travel and service times in vehicle routing. *Transp Sci* 50(2):627–641. <https://doi.org/10.1287/trsc.2015.0601>
- Haas CN (1999) On modeling correlated random variables in risk assessment. *Risk Anal* 19(6):1205–1214. <https://doi.org/10.1023/A:1007047014741>
- Hall RW (1986) The fastest path through a network with random time-dependent travel times. *Transp Sci* 20(3):182–188. <https://doi.org/10.1287/trsc.20.3.182>
- Handler GY, Zang I (1980) A dual algorithm for the constrained shortest path problem. *Networks* 10(4):293–309. <https://doi.org/10.1002/net.3230100403>
- Huang H, Gao S (2012) Optimal paths in dynamic networks with dependent random link travel times. *Transp Res Part B Methodol* 46(5):579–598
- Ji Z, Kim YS, Chen A (2011) Multi-objective α -reliable path finding in stochastic networks with correlated link costs: a simulation-based multi-objective genetic algorithm approach (SMOGA). *Expert Syst Appl* 38(3):1515–1528. <https://doi.org/10.1016/j.eswa.2010.07.064>
- Joksch HC (1966) The shortest route problem with constraints. *J Math Anal Appl* 14(2):191–197. [https://doi.org/10.1016/0022-247x\(66\)90020-5](https://doi.org/10.1016/0022-247x(66)90020-5)
- Latouche G, Ramaswami V (1999) Introduction to matrix analytic methods in stochastic modeling. *Soc Ind Appl Math*. doi 10(1137/1):9780898719734
- Li J, Han X (2019) Revised pulse algorithm for elementary shortest path problem with resource constraints. In: 2019 seventh international symposium on computing and networking (CANDAR), pp 37–44. <https://doi.org/10.1109/CANDAR.2019.00013>
- Li ST, Hammond JL (1975) Generation of pseudorandom numbers with specified univariate distributions and correlation coefficients. *IEEE Trans Syst Man Cybern SMC* 5(5):557–561
- Li X, Tian P, Leung SCH (2010) Vehicle routing problems with time windows and stochastic travel and service times: models and algorithm. *Int J Prod Econ* 125(1):137–145. <https://doi.org/10.1016/j.ijpe.2010.01.013>
- Lozano L, Medaglia AL (2013) On an exact method for the constrained shortest path problem. *Comput Oper Res* 40(1):378–384. <https://doi.org/10.1016/j.cor.2012.07.008>
- Lozano L, Duque D, Medaglia AL (2016) An exact algorithm for the elementary shortest path problem with resource constraints. *Transp Sci* 50(1):348–357. <https://doi.org/10.1287/trsc.2014.0582>
- Lurie PM, Goldberg MS (1998) An approximate method for sampling correlated random variables from partially-specified distributions. *Manag Sci* 44(2):203–218
- Miller-Hooks E (2001) Adaptive least-expected time paths in stochastic, time-varying transportation and data networks. *Networks* 37(1):35–52. 10.1002/1097-0037(200101)37:1<35::AID-NET4>3.0.CO;2-G
- Miller-Hooks E, Mahmassani H (2003) Path comparisons for a priori and time-adaptive decisions in stochastic, time-varying networks. *Eur J Oper Res* 146(1):67–82. [https://doi.org/10.1016/S0377-2217\(02\)00231-X](https://doi.org/10.1016/S0377-2217(02)00231-X)
- Miller-Hooks ED, Mahmassani HS (1998) Least possible time paths in stochastic, time-varying networks. *Comput Oper Res* 25(12):1107–1125. [https://doi.org/10.1016/S0305-0548\(98\)00027-6](https://doi.org/10.1016/S0305-0548(98)00027-6)
- Miller-Hooks ED, Mahmassani HS (2000) Least expected time paths in stochastic, time-varying transportation networks. *Transp Sci* 34(2):198–215. <https://doi.org/10.1287/trsc.34.2.198.12304>
- Mirchandani PB (1976) Shortest distance and reliability of probabilistic networks. *Comput Oper Res* 3(4):347–355. [https://doi.org/10.1016/0305-0548\(76\)90017-4](https://doi.org/10.1016/0305-0548(76)90017-4)
- Nie YM, Wu X (2009) Shortest path problem considering on-time arrival probability. *Transp Res Part B Methodol* 43(6):597–613. <https://doi.org/10.1016/j.trb.2009.01.008>
- Perez JF, Silva DF, Goez JC, Sarmiento A, Akhavan Tabatabaei R, Riano G (2017) Algorithm 972: jMarkov: an integrated framework for Markov chain modeling. In: ACM Transactions on Mathematical Software. <https://github.com/coin-or/jMarkov>. Accessed 1 Mar 2021
- Prakash AA (2018) Pruning algorithm for the least expected travel time path on stochastic and time-dependent networks. *Transp Res Part B Methodol* 108:127–147. <https://doi.org/10.1016/j.trb.2017.12.015>

- Santos L, Coutinho-Rodrigues J, Current JR (2007) An improved solution algorithm for the constrained shortest path problem. *Transp Res Part B Methodol* 41(7):756–771. <https://doi.org/10.1016/j.trb.2006.12.001>
- Sedeño-Noda A, Alonso-Rodríguez S (2015) An enhanced K-SP algorithm with pruning strategies to solve the constrained shortest path problem. *Appl Math Comput* 265:602–618. <https://doi.org/10.1016/j.amc.2015.05.109>
- Sen S, Pillai R, Joshi S, Rathi AK (2001) A mean-variance model for route guidance in advanced traveler information systems. *Transp Sci* 35(1):37–49. <https://doi.org/10.1287/trsc.35.1.37.10141>
- Shen L, Shao H, Wu T, Lam WH, Zhu EC (2019) An energy-efficient reliable path finding algorithm for stochastic road networks with electric vehicles. *Transp Res Part C Emerg Technol* 102(March):450–473. <https://doi.org/10.1016/j.trc.2019.03.020>
- Sigal CE, Pritsker AAB, Solberg JJ (1980) The stochastic shortest route problem. *Oper Res* 28(5):1122–1129. <https://doi.org/10.1287/opre.28.5.1122>
- Sivakumar RA, Batta R (1994) The variance-constrained shortest path problem. *Transp Sci* 28(4):309–316. <https://doi.org/10.1287/trsc.28.4.309>
- Stabler B (2016) Transportation networks for research. In: GitHub repository. <https://github.com/bstabler/TransportationNetworks>. Accessed 1 Mar 2021
- Thomas BW, Calogiuri T, Hewitt M (2019) An exact bidirectional A* approach for solving resource-constrained shortest path problems. *Networks* 73(2):187–205. <https://doi.org/10.1002/net.21856>
- Thummler A, Buchholz P, Telek M (2006) A novel approach for phase-type fitting with the EM algorithm. *IEEE Trans Depend Secure Comput* 3(3):245–258. <https://doi.org/10.1109/tdsc.2006.27>
- Wang L, Yang L, Gao Z (2016) The constrained shortest path problem with stochastic correlated link travel times. *Eur J Oper Res* 255(1):43–57. <https://doi.org/10.1016/j.ejor.2016.05.040>
- Xing T, Zhou X (2011) Finding the most reliable path with and without link travel time correlation: a Lagrangian substitution based approach. *Transp Res Part B Methodol* 45(10):1660–1679. <https://doi.org/10.1016/j.trb.2011.06.004>
- Yang L, Yang X, You C (2013) Stochastic scenario-based time-stage optimization model for the least expected time shortest path problem. *Int J Uncertainty Fuzziness Knowl Based Syst* 21(SUPPL.1):17–33
- Zang Z, Xu X, Yang C, Chen A (2018) A distribution-fitting-free approach to calculating travel time reliability ratio. *Transp Res Part C Emerg Technol* 89:83–95. <https://doi.org/10.1016/j.trc.2018.01.027>
- Zeng W, Miwa T, Wakita Y, Morikawa T (2015) Application of Lagrangian relaxation approach to α -reliable path finding in stochastic networks with correlated link travel times. *Transp Res Part C Emerg Technol* 56:309–334. <https://doi.org/10.1016/j.trc.2015.04.018>
- Zhang Y (2019) An algorithm for reliable shortest path problem with travel time correlations. Ph.D. thesis. <https://doi.org/10.1037/0033-2909.126.1.78>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.