Transportation Science

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

On Modeling Stochastic Travel and Service Times in Vehicle Routing

Andrés Gómez, Ricardo Mariño, Raha Akhavan-Tabatabaei, Andrés L. Medaglia, Jorge E. Mendoza

TRANSPORTATION SCIENCE
Vol. 50, No. 2, May 2016, pp. 627–641
ISSN 0041-1655 (print) | ISSN 1526-5447 (online)

http://dx.doi.org/10.1287/trsc.2015.0601

On Modeling Stochastic Travel and Service Times in Vehicle Routing

Andrés Gómez, Ricardo Mariño, Raha Akhavan-Tabatabaei, Andrés L. Medaglia
Centro para la Optimización y Probabilidad Aplicada (COPA), Departamento de Ingeniería Industrial,
Universidad de los Andes, Bogotá, Colombia
{a.gomez70@uniandes.edu.co, ricardof1c@hotmail.com, r.akhavan@uniandes.edu.co, amedagli@uniandes.edu.co}

Jorge E. Mendoza
Université Catholique de l'Ouest, LARIS (EA7315), 49008 Angers, France; and Université François-Rabelais de Tours,
CNRS, LI (EA6300), OC (ERL CNRS 6305), 37200 Tours, France, jorge.mendoza@univ-tours.fr

Vehicle routing problems with stochastic travel and service times (VRPSTT) consist of designing transportation routes of minimal expected cost over a network where travel and service times are represented by random variables. Most of the existing approaches for VRPSTT are conceived to exploit the properties of the distributions assumed for the random variables. Therefore, these methods are tied to a given family of distributions and subject to strong modeling assumptions. We propose an alternative way to model travel and service times in VRPSTT without making many assumptions regarding such distributions. To illustrate our approach, we embed it into a state-of-the-art routing engine and use it to conduct experiments on instances with different travel and service time distributions.

Keywords: vehicle routing; stochastic travel times; stochastic service times; Phase-type distributions
History: Received: May 2013; revisions received: March 2014, October 2014; accepted: January 2015. Published online in *Articles in Advance* July 14, 2015.

1. Introduction

Vehicle routing problems (VRPs) are concerned with the design of efficient routes that deliver goods and services from (to) central depots to (from) customer locations, satisfying specific business constraints. Since the 1960s, a vast amount of research has been devoted to solve different VRP variants. Most of the solution methods for VRPs are based on the premise that problem parameters such as travel, service times, and customer demands are known in advance. However, in a practical setting, more often than not, the problem parameters (customers, demands, or travel times) are uncertain, and neglecting their stochastic nature may lead to poor routing decisions. Each uncertain component poses unique challenges that result in different solution approaches (Gendreau, Jabali, and Rei 2014). In this research we address the family of problems that has received less attention in the literature: the VRPs with stochastic travel and service times (VRPSTTs).

The family of VRPSTTs is defined on a complete graph $G=(\mathscr{V},\mathscr{E})$, where $\mathscr{V}=\{0,v_1,v_2,\ldots,v_n\}$ is the vertex set and $\mathscr{E}=\{e=(v_i,v_j)\colon v_i,v_j\in\mathscr{V}, v_i\neq v_j\}$ is the edge set. Vertices $v_i\in\mathscr{V}\backslash\{0\}$ represent the customers and vertex 0 represents the depot. An edge weight $\tilde{t}_e$, associated with edge $e=(v_i,v_j)$, represents the random travel time along edge e. Each customer $v_i\in V\backslash\{0\}$ has a random service time $\tilde{s}_{v_i}$ and a known demand d_{v_i} for a given product. Both travel and service times are assumed to follow known distributions. Customers are served by an unlimited fleet of homogeneous vehicles located at the depot, each with a maximum capacity Q.

The objective is to design a route set $\mathscr{R}$ of minimum total expected duration $E[\tilde{T}(\mathscr{R})]=\sum_{r\in\mathscr{R}}E[\tilde{T}_r]$, where $\tilde{T}(\mathscr{R})$ is the total (random) duration of the route set, $\tilde{T}_r$ is the (random) duration of route r, and $E[\cdot]$ denotes the expected value. Each route $r\in\mathscr{R}$ is a tuple $r=(0,v_{(1)},\ldots,v_{(i)},\ldots,v_{(n_r)},0)$, where $v_{(i)}\in\mathscr{V}\backslash\{0\}$ is the ith customer visited in the route, n_r is the number of customers serviced by the route, and $(v_{(i)},v_{(i+1)})\in\mathscr{E}$ (with $v_{(0)}=v_{(n_r+1)}=0$). We will refer to route r, depending on the context, either as the sequence of vertices or edges in the route. Aside from the classical capacity constraint, each route $r\in\mathscr{R}$ satisfies a set of constraints $\mathscr{C}$ that involve the route duration. For instance, set $\mathscr{C}$ may include not exceeding a time limit T or violating customer time windows.

Search algorithms for VRPSTT can be decoupled into two components: the *optimization engine* and the *route evaluator*. The optimization engine is responsible for exploring the solution space, unveiling new routes to make up a solution. The vehicle routing literature comprises a number of efficient search algorithms that, with mild adaptations, can play the role of

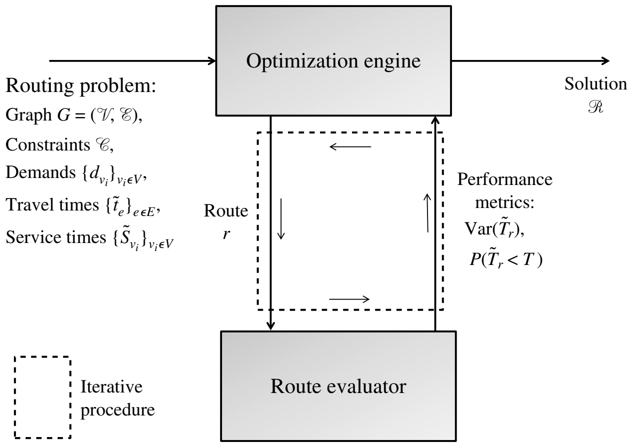

Figure 1 VRPSTT Algorithm Description

the optimization engine. These include genetic algorithms, tabu search, and large neighborhood search, among others (Gendreau, Laporte, and Potvin 2001; Laporte 2007; Doerner and Schmid 2010). The route evaluator, in contrast, is responsible for extracting the *performance metrics* of a route such as the travel time variance, $\mathrm{Var}(\tilde{T}_r)$, or the probability of completing the route before the threshold T, $P(\tilde{T}_r < T)$. Figure 1 depicts the interaction between these two components.

Note that many performance metrics involving the total duration $\tilde{T}_r$ of route r require the computation of the convolution of several random variables that represent travel and service times. Depending on the probability distributions assumed for the random variables, computing these convolutions can be a straightforward procedure (e.g., normal distributions) or a complex operation (e.g., lognormal distributions). Choosing a distribution to model travel and service times is thus a central decision when modeling and solving VRPSTT.

The first concern when selecting a distribution is its ability to *accurately* model travel and service times. Most researchers agree that vehicle travel times' variance and skewness increase in interrupted traffic or in the presence of incidents or accidents. Besides these general guidelines, there appears to be no consensus on which is the most accurate distribution for modeling travel times. In fact, such a distribution may not exist at all, and the distributions should be hand-picked specifically for each problem or even for each edge. For instance, Lecluyse, Van Woensel, and Peremans (2009) argue that a lognormal distribution accurately reflects travel times under uninterrupted traffic: there is a minimum time needed to cover the distance, after which the probability increases rapidly to a maximum and then slowly decreases with a long tail. Boyles, Voruganti, and Waller (2010) define environmental states and a probability distribution over the states and associate each state with a distribution function. They model travel times using the mixture distribution obtained as the convex combination of the distributions associated with each state. In a more recent study, Susilawati, Taylor, and Somenahalli (2013) used data from urban corridors (i.e., interrupted traffic) to test the normal, lognormal, gamma, Weibull, generalized Pareto, and Burr distributions. They found that although the best distribution depends on the particular link under consideration, in general the Burr and generalized Pareto distributions, which have large or even infinite variances, fit the data better than the other distributions. Moreover, they noted that short links had in some cases longer tails due to congestion, and in other cases, the links showed bimodality (which they attributed to the impact of traffic lights) and could not be accurately described by any of the tested distributions.[1]

A second concern when selecting distributions is the computational burden. For this reason, the VRPSTT literature has focused mainly on cases where travel and service times follow a single additive family of distributions. In this scenario, computing the convolution of random variables can be done efficiently. For instance, in their pioneering VRPSTT work Laporte, Louveaux, and Mercure (1992) assume that the travel times are independent and normally distributed random variables and exploit the fact that the convolution of normal random variables is again normally distributed. A similar assumption is made by Kenyon and Morton (2003), Figliozzi (2010), and Chen et al. (2014). Taş et al. (2013, 2014) model travel times using the family of gamma distributions, which is also an additive family.

Dealing with nonadditive probability distributions is not trivial, as computing the convolutions of the travel times and service times becomes a daunting task. However, different approaches that circumvent this difficulty have been proposed in the literature. On one hand, Van Woensel et al. (2003, 2007) use the vehicle flow in a road to determine the arrival rate to a fictitious queueing system whose service rate is obtained from the free flow speed at the same road. Using this approach, the authors map each road's specific characteristics to the parameters of the queueing system and then obtain the expected value and variance of the travel time across each road. On the other hand, Kenyon and Morton (2003) and Li, Tian, and Leung (2010) use Monte Carlo simulations to model the stochastic times, and Errico et al. (2013) assume that service times are discrete with finite support. Such methods can model a wide array of different distributions, but they implicitly assume a maximum value

[1] Bimodal distributions can be represented as the mixture of two distributions, describing the free flow and congested travel times.

for the distributions and may have trouble representing the extreme values of heavy-tailed distributions.

Modeling stochastic travel and service times with the aforementioned limitations poses several difficulties when tackling VRPSTT. First, the ability to develop a general solution strategy that can be applied to different real-life scenarios is hampered by the lack of a universally acceptable distribution for modeling travel and service times. Second, finding a single distribution that can accurately model all of the travel and service times in an instance may be impossible. Third, computing the convolutions of the chosen distributions is in general not a trivial task, especially if the travel or service times follow different families of distributions.

To overcome these difficulties we propose a new approach using Phase-type (PH) distributions to model the stochastic travel and service times. Using this family of distributions, we can closely approximate any positive, continuous distribution with precision and compute the convolutions in an exact manner. To assess the benefits of our approach, we implemented the proposed PH route evaluator as well as route evaluators based on normal distributions and Monte Carlo simulation. We embedded the route evaluators into an adaptation of the multispace sampling heuristic proposed by Mendoza and Villegas (2013). We use the resulting algorithms to compare the three approaches in terms of both solution quality and computational performance.

The remainder of this paper is organized as follows. Section 2 discusses the modeling of stochastic travel and service times using PH distributions. Section 3 describes the multispace sampling heuristic and its components. Section 4 discusses the computational results over a set of instances with stochastic travel and service times. Section 5 concludes the paper.

2. Modeling Stochastic Travel and Service Times with Phase-Type Distributions

To the best of our knowledge, PH distributions have never been used in the context of VRPSTT. However, their density and closure properties make them an attractive choice for modeling vehicle travel and service times in real-life scenarios. This section introduces PH distributions and their properties and illustrates their application in the context of stochastic vehicle routing.

2.1. Definition and Properties

Neuts (1981) introduced PH distributions, generalizing Erlang's idea of method of stages, where random time intervals are modeled as being composed of a number of exponentially distributed segments. Consider a continuous time Markov chain

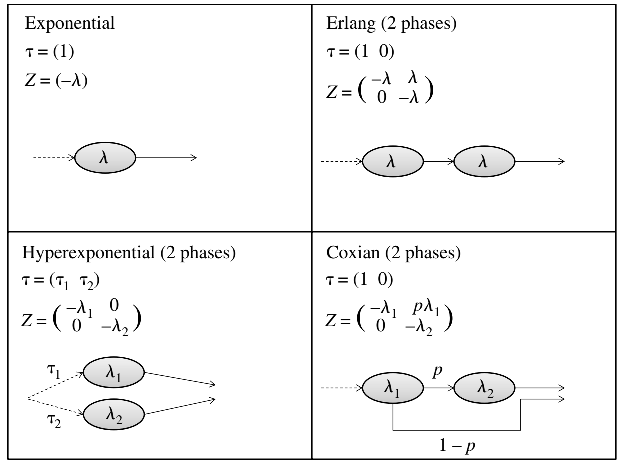

Figure 2 Common PH Distributions

(CTMC) $\{X(t)\}_{t>0}$ defined on the finite state space $S = \{1, 2, \ldots, m, 0\}$, where the first m states (phases) are transient and the state 0 is an absorbing state. The initial probability vector of the Markov chain is $(\boldsymbol{\tau}, \tau_0)$ and the infinitesimal generator (see Online Appendix A (available as supplemental material at http://dx.doi.org/10.1287/trsc.2015.0601)) is

$$U = \begin{pmatrix} \mathbf{Z} & \mathbf{z} \\ \mathbf{0} & 0 \end{pmatrix}, \tag{1}$$

where $\boldsymbol{\tau}$ is a row vector of size m containing the probability of beginning in each of the transient states, τ_0 is the probability of beginning in the absorbing state, $\mathbf{z}$ is a column vector of size m representing the transition rates into the absorbing state, and $\mathbf{Z}$ is an $m \times m$ matrix representing the transitions between the transient states. Because U is the generator of a CTMC, $Z_{ii} \le 0$, $Z_{ij} \ge 0$ for $i \ne j$, $z_i \ge 0$, and $\mathbf{Z} \cdot \mathbf{1} + \mathbf{z} = 0$, where $\mathbf{1}$ is a column vector of 1s. We say that Y is distributed $\text{PH}(\boldsymbol{\tau}, \mathbf{Z})$ if Y is the time until absorption into state 0 in this CTMC (Latouche and Ramaswami 1999). Figure 2 gives matricial and graphical representations of some of the simplest PH distributions, including the exponential and Erlang distributions.

PH distributions are dense in the set of continuous density functions with support on $[0, \infty)$, meaning that there exists a PH distribution arbitrarily close to any positive distribution. Finding such a PH distribution is a process known as distribution fitting. For that process, different efficient algorithms exist, like those proposed by Telek and Heindl (2002), Bobbio, Horvát, and Telek (2005), and Thummler, Buchholz, and Telek (2006).

PH distributions have properties that make them flexible and algorithmically tractable (Neuts 1981). First, the distribution function, the expected value, and higher-order moments of a PH distribution can be found in closed formulae in terms of $\boldsymbol{\tau}$ and $\mathbf{Z}$. Second, the convolution of PH distributions is again a PH distribution where the initial vector and generator matrix are easily obtained from the original parameters of the distributions. Formal statements of these properties can be found in Online Appendix B.

The density and closure properties of PH distributions make them an attractive choice for modeling vehicle travel and service times. On one hand, since travel and service times are always positive, theoretically they can be modeled as accurately as needed using PH distributions. The extensive (and ongoing) research on PH distribution fitting provides a practical mechanism to find an adequate approximation, as long as there is sufficient empirical data. On the other hand, regardless of the PH distribution used for each travel or service time, the convolution is obtained using the same formula (Online Appendix B). PH distributions are thus able to accurately model different travel or service times while allowing the computation of the required convolutions in an efficient manner.

2.2. Example: Fitting a PH Distribution to Real Traffic Data

In practice, travel time distributions are often drawn from empirical data, so their probability density function is not known explicitly. In fact, closed formulae for the probability density function may not exist at all. Moreover, travel times have a minimum value that corresponds to the time required to travel the distance at free flow speed. For that reason, it is difficult to work with distributions that do not admit a minimum, like the normal distribution. In contrast, PH distributions provide a good alternative to use under these conditions. We illustrate their applicability by fitting a PH distribution to a travel time data set.

Our data set (available online at http://hdl.handle.net/1992/1179) consists of 203 commuting times between the town of La Calera (Cundinamarca, Colombia) and downtown Bogotá (Cundinamarca, Colombia). The expected travel time is 54 minutes, the standard deviation is 12 minutes, and the minimum and maximum travel times are 39 and 120 minutes, respectively. To account for the minimum travel time, we use a *shifted* distribution $\bar{Y} = y_0 + Y$, where y_0 is the minimum travel time and Y is a PH distribution. Using the Expectation Maximization algorithm proposed by Thummler, Buchholz, and Telek (2006), we have that Y is $PH(\boldsymbol{\tau}, \mathbf{Z})$ with

$$\boldsymbol{\tau} = (0.25 \quad 0.00 \quad 0.00 \quad 0.75 \quad 0.00 \quad 0.00),$$

$$\mathbf{Z} = \begin{pmatrix} -0.11 & 0.11 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & -0.11 & 0.11 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & -0.11 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & -0.27 & 0.27 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & -0.27 & 0.27 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & -0.27 \end{pmatrix}.$$

Figures 3 and 4 show the probability density functions and cumulative distribution functions of the original data set, the fitted PH distribution $\bar{Y}$, and the normal

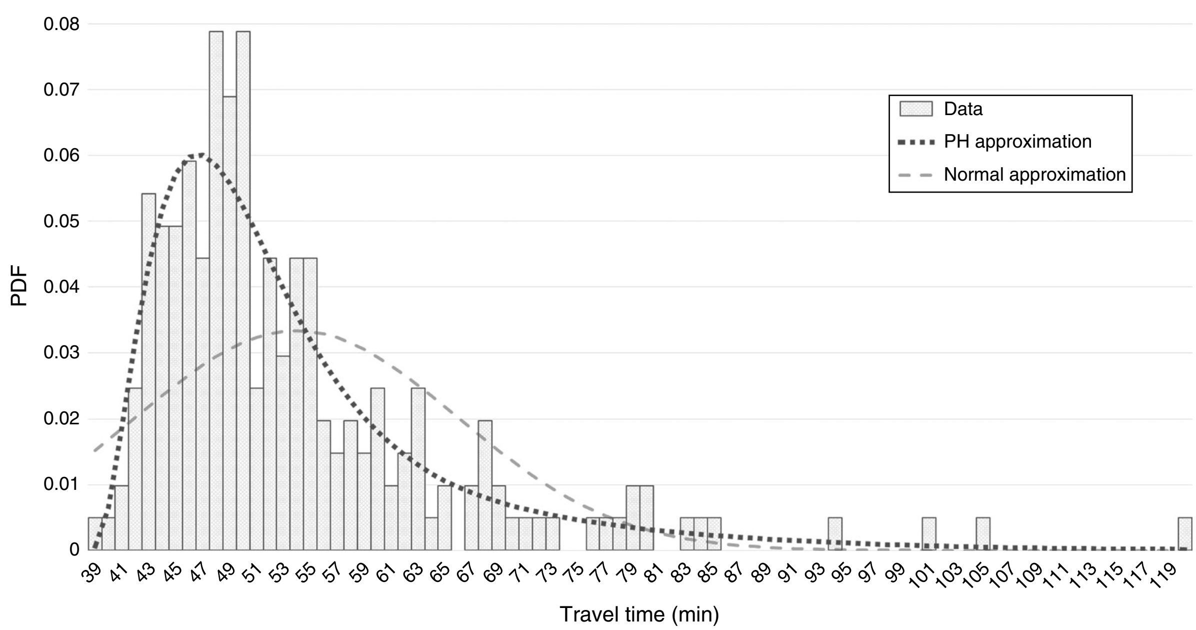

Figure 3 Probability Density Functions of the Original Data Set and the Approximated Distributions

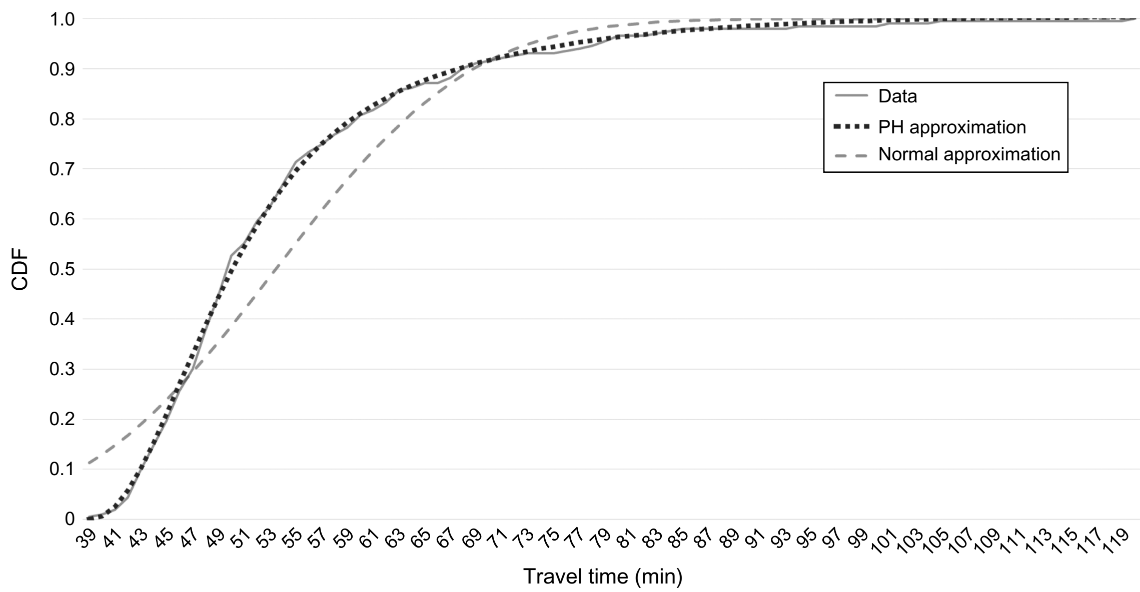

Figure 4 **Cumulative Distribution Functions of the Original Data Set and the Approximated Distributions**

distribution that matches the first two moments of the data set (the normal distribution is the most broadly used distribution in the VRPSTT literature). In this case, the PH approximation has a mean squared error of 0.76 and is a better fit than the normal approximation, which has a mean squared error of 2.72.

Note that performance metrics related to the shifted distribution $\bar{Y} = y_0 + Y$ can easily be inferred from Y as follows:

1. $E[\bar{Y}] = y_0 + E[Y]$
2. $\mathrm{Var}(\bar{Y}) = \mathrm{Var}(Y)$
3. $P(\bar{Y} < T) = P(Y < T - y_0)$
4. If $\bar{Y}_i = y_{0_i} + Y_i$, $i = 1, 2$, then $\bar{Y}_1 * \bar{Y}_2 = y_{0_1} + y_{0_2} + Y_1 * Y_2$, where $*$ denotes the convolution operator.

Therefore, without loss of generality, throughout the rest of the paper we will assume that distributions used to model travel and service times are not shifted.

2.3. Example: Modeling a Route's Total Duration

Figure 5 depicts route $r = (0, 1, 2, 0)$ serving two customers. In this instance, travel and service times are stochastic. We want to evaluate the expected total duration and its variance, as well as the probability that the duration does not exceed a given threshold T.

Suppose that, as reported in Boyles, Voruganti, and Waller (2010), for the travel time $\tilde{t}_{12}$ three environmental states have been identified: the first state happens with probability p_1 and corresponds to good weather, in which case the travel time has low variance and is close to the free flow speed; the second and third states happen with probabilities p_2 and p_3 and correspond to bad weather and incident, respectively, in which case the expected travel times are longer and have high variance as well. Let us use an Erlang distribution to model low-variance travel times and exponential distributions to model high-variance travel times. Then $\tilde{t}_{12}$ follows a PH distribution $\mathcal{D}$ with the structure shown in Figure 6: the travel time consists of two exponential phases with probability p_1, or one exponential phase with probability p_2 or p_3 with different exponential rates, depending on the environmental state.

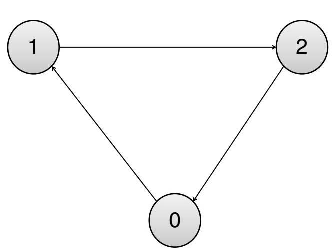

Figure 5 **Graph Representation of Route** *r*

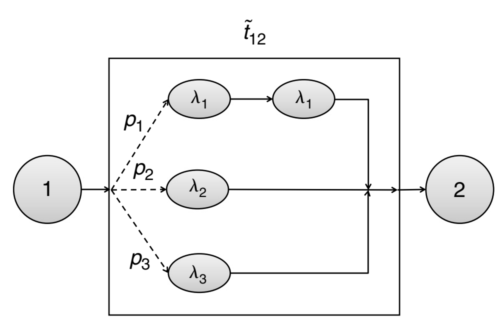

Figure 6 **Structure of PH Distribution** $\mathcal{D}$

Let $p_1 = 0.68$, $p_2 = 0.31$, $p_3 = 0.01$, $\lambda_1 = 0.30$, $\lambda_2 = 0.05$, and $\lambda_3 = 0.01$. Then $\tilde{t}_{1,2}$ is $PH(\boldsymbol{\tau}_{1,2}, \mathbf{Z}_{1,2})$ with

$$\boldsymbol{\tau}_{1,2} = (0.68 \quad 0.00 \quad 0.31 \quad 0.01),$$

$$\mathbf{Z}_{1,2} = \begin{pmatrix} -0.30 & 0.30 & 0.00 & 0.00 \\ 0.00 & -0.30 & 0.00 & 0.00 \\ 0.00 & 0.00 & -0.05 & 0.00 \\ 0.00 & 0.00 & 0.00 & -0.01 \end{pmatrix}.$$

Now suppose $\tilde{t}_{0,1}$ follows an exponential distribution with rate 0.1. Then $\tilde{t}_{0,1}$ is $PH(\boldsymbol{\tau}_{0,1}, \mathbf{Z}_{0,1})$ with

$$\boldsymbol{\tau}_{0,1} = (1.0),$$
$$\mathbf{Z}_{0,1} = (-0.1).$$

Moreover, suppose $\tilde{t}_{2,0}$ follows an Erlang distribution with two phases and rate 0.4. Then $\tilde{t}_{2,0}$ is $PH(\boldsymbol{\tau}_{2,0}, \mathbf{Z}_{2,0})$ with

$$\boldsymbol{\tau}_{2,0} = \begin{pmatrix} 1.0 & 0.0 \end{pmatrix},$$

$$\mathbf{Z}_{2,0} = \begin{pmatrix} -0.4 & 0.4 \\ 0.0 & -0.4 \end{pmatrix}.$$

Finally, suppose $\tilde{s}_{v_1}$ (i.e., the service time at customer 1) is exponentially distributed with rate 0.2, and $\tilde{s}_{v_2}$ is $PH(\boldsymbol{\tau}_{v_2}, \mathbf{Z}_{v_2})$ with

$$\boldsymbol{\tau}_{v_2} = (0.5 \quad 0.5),$$

$$\mathbf{Z}_{v_2} = \begin{pmatrix} -0.1 & 0.0 \\ 0.0 & -0.5 \end{pmatrix}.$$

Figure 7 depicts a representation of the PH distributions of each travel and service time. The total duration of route r, $\tilde{T}_r$, is the convolution of all of the PH distributions shown in Figure 7, which is distributed $PH(\boldsymbol{\tau}_r, \mathbf{Z}_r)$ with

$$\boldsymbol{\tau}_r = (1.00\ 0.00\ 0.00\ 0.00\ 0.00\ 0.00\ 0.00\ 0.00\ 0.00\ 0.00)$$

$$\mathbf{Z}_r = \begin{pmatrix} -0.10 & 0.10 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & -0.20 & 0.14 & 0.00 & 0.06 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & -0.30 & 0.30 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & -0.30 & 0.00 & 0.00 & 0.15 & 0.15 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & -0.05 & 0.00 & 0.03 & 0.03 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & -0.01 & 0.01 & 0.01 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & -0.10 & 0.00 & 0.10 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & -0.50 & 0.50 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & -0.40 & 0.40 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & -0.40 \end{pmatrix}.$$

Note that the diagonal entries $(\mathbf{Z}_r)_{ii}$ correspond to the negative of the rate of the corresponding phase, and $(\mathbf{Z}_r)_{ij}$, $i \neq j$, is the product of the rate of phase i and the transition probability to phase j. For example, in the second row, $(\mathbf{Z}_r)_{22}$ is the negative of the rate of the exponential service time at customer 1, $(\mathbf{Z}_r)_{23} = 0.2 \times 0.68 = 0.136$ (we round up to 0.14 for the sake of brevity), $(\mathbf{Z}_r)_{25} = 0.2 \times 0.31 = 0.062$, and $(\mathbf{Z}_r)_{26} = 0.2 \times 0.01 = 0.002$. Figure 8 shows a representation of $PH(\boldsymbol{\tau}_r, \mathbf{Z}_r)$.

Using the formula introduced in Online Appendix B, we can compute the required performance measures from $PH(\boldsymbol{\tau}_r, \mathbf{Z}_r)$. The expected duration of route r is $E[\tilde{T}_r] = 37.73$, its variance is $\mathrm{Var}(\tilde{T}_r) = 561.16$, the probability that the route is completed before 50 units of time is $P(\tilde{T}_r \leq 50) = 0.80$, and the probability that it is completed before 75 units of time is $P(\tilde{T}_r \leq 75) = 0.95$.

3. The Optimization Engine

Using the concepts introduced in §2, we can build a black box $\mathtt{PH}(r)$ that takes as input a route r and returns PH_r, the PH distribution of the total duration of route r. The black box $\mathtt{PH}(r)$ can be embedded as a route evaluator within different optimization algorithms for VRPSTT.

To illustrate the advantages of PH distributions for modeling travel and service times, we compare the performance of $\mathtt{PH}(r)$ against alternative route evaluators. To conduct our experiments, we embedded the different route evaluators in the same routing engine. For that purpose we implemented an algorithm based on the multispace sampling heuristic (MSH) by Mendoza and Villegas (2013). We selected MSH as the optimization engine because of its convenience for our comparative study. Indeed, as opposed to local search-based algorithms, MSH samples the solution space rather than following trajectories within the space by iteratively changing solutions. In other words, MSH explores the same areas of the solution space regardless of the route evaluator being used. For the sake of completeness, in the following subsections we present a description of MSH.

3.1. Multispace Sampling Heuristic Overview

The MSH follows a two-phase solution strategy. In the first phase, it samples multiple solution representation spaces; in the second it assembles the best possible solution using parts of the sampled elements. The approach operates as follows. At each iteration k, the algorithm selects a *sampling heuristic* from a set $\mathcal{H}$ of randomized traveling salesman problem (TSP) heuristics and uses it to build a giant tour p^k visiting all customers. Then the algorithm makes a call to a *splitting procedure*, similar to the one described in Prins (2004), to retrieve a tuple $\langle \Omega^k, s^k \rangle$, where Ω^k is the set of all feasible routes (in terms of not exceeding the vehicle capacity Q and satisfying the constraints in $\mathscr{C}$) that can be extracted from p^k without altering the order of the customers, and s^k is one of the best solutions that can be built using routes from Ω^k. The

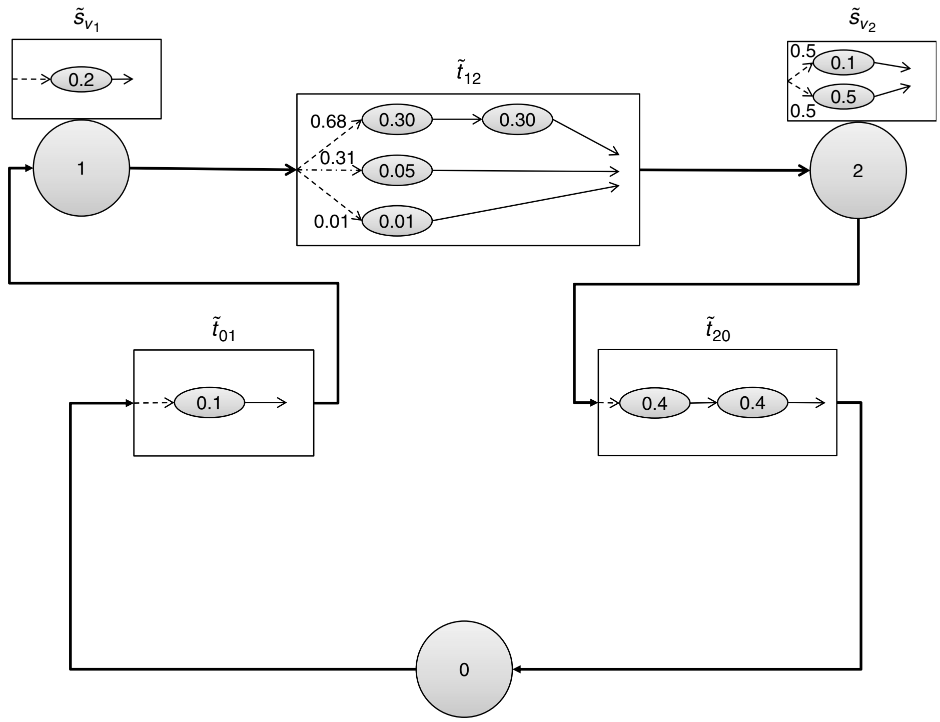

Figure 7 PH Distributions of Travel and Service Times

routes in Ω^k join a set of sampled routes Ω (i.e., $\Omega \leftarrow \Omega \cup \Omega^k$), and s^k is used to update an upper bound $f(s^*)$ on the objective function of the final solution, where s^* is the best solution found so far. After a total of K iterations, the heuristic proceeds to the assembly phase, which consists of solving a set partitioning (column-oriented) formulation of the underlying routing problem over Ω, using $f(s^*)$ as an upper bound. A pseudocode of MSH is given in Algorithm 1. Figure 9 illustrates how the algorithm operates, showing an example with $K = 2$ iterations.

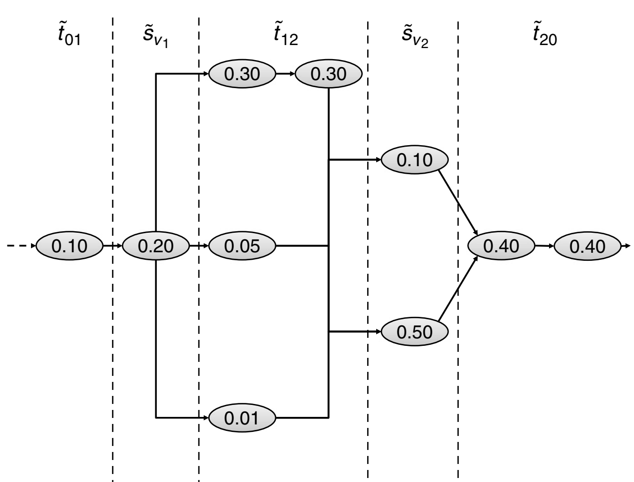

Figure 8 Structure of $PH(\boldsymbol{\tau}_r, \mathbf{Z}_r)$

Algorithm 1 (Multispace sampling)
Input: G, VRP graph; Q, vehicle capacity; $\mathscr{C}$, set of duration-related constraints; $\mathscr{H}$, set of TSP heuristics; K, maximum number of iterations.
Output: $\mathscr{R}$, best set of routes found.

$\Omega \leftarrow \varnothing$
$k \leftarrow 1$
while $k \leq K$ **do**
    $h \leftarrow$ `selectHeuristic`$(\mathscr{H})$
    $p^k \leftarrow h(G)$
    $\langle \Omega^k, s^k \rangle \leftarrow$ `splittingProcedure`$(G, Q, \mathscr{C}, p^k)$
    $\Omega \leftarrow \Omega \cup \Omega^k$
    if $k = 1$, **then**
        $s^* \leftarrow s^k$
    else if $f(s^k) \leq f(s^*)$, **then**
        $s^* \leftarrow s^k$
    end if
    $k \leftarrow k + 1$
end while
$\mathscr{R} \leftarrow$ `setPartitioningProblem`(G, Ω, s^*)
return $\mathscr{R}$

The remainder of this section describes the three main components of the heuristic, namely, the TSP heuristics, the splitting procedure, and the set partitioning problem in the assembly phase.

3.2. Traveling Salesman Problem Heuristics

Set $\mathscr{H}$ contains several randomized constructive TSP heuristics. In our experiments the algorithm uses the same heuristics described by Mendoza and Villegas

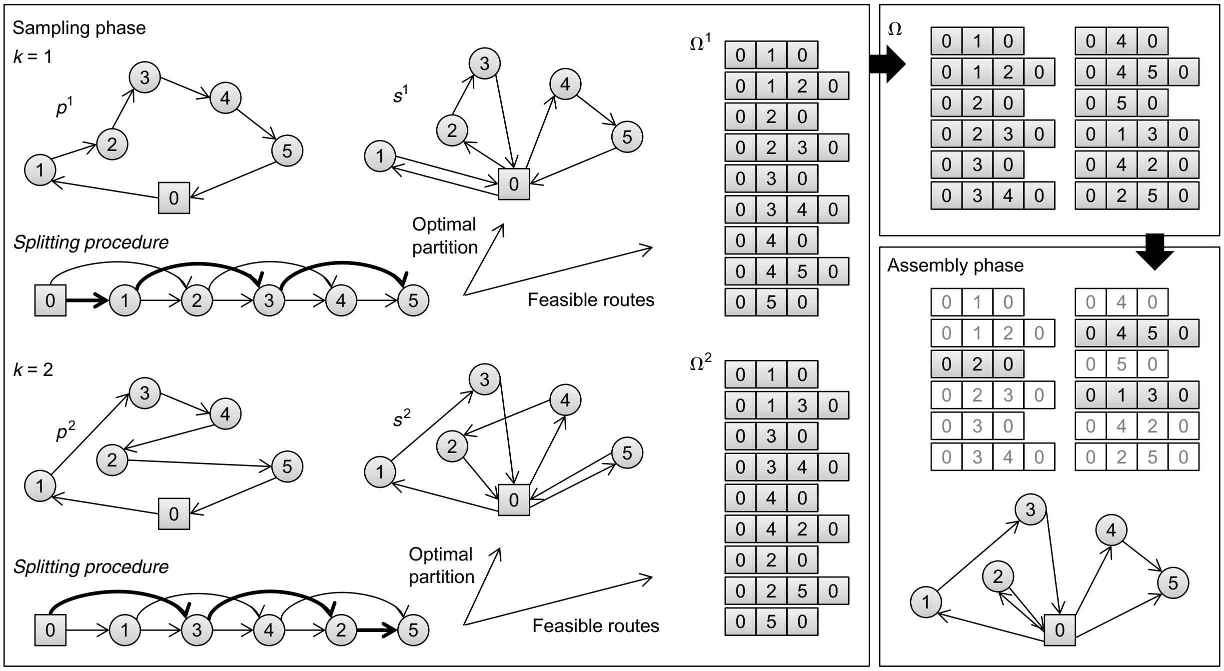

Figure 9 **MSH Example with $K = 2$**

(2013): randomized nearest neighbor (RNN), randomized nearest insertion (RNI), randomized farthest insertion (RFI), and randomized best insertion (RBI). For the sake of completeness, we present them in this section.

Let $p_0^k = (v_{(0)}, v_{(1)}, \ldots, v_{(n_p)})$ be the partial TSP tour being built by a given sampling heuristic at iteration k and $\mathscr{W}$ the set of unrouted vertices (yet to be sequenced in the TSP tour p_0^k). Define $t_{(v,w)} = E[\tilde{t}_{(v,w)}]$ as the known expected travel time along edge $(v, w) \in \mathscr{E}$, and for all $w \in \mathscr{W}$ define the following three metrics:

$$t_{\min}(w) = \min_{j=0,\ldots,n_p} t_{(v_{(j)}, w)},$$

$$t_{\max}(w) = \max_{j=0,\ldots,n_p} t_{(v_{(j)}, w)},$$

$$\Lambda_{\min}(w) = \min_{j=0,\ldots,n_p-1} \{t_{(v_{(j)}, w)} + t_{(w, v_{(j+1)})} - t_{(v_{(j)}, v_{(j+1)})}\}.$$

Finally, let x be a random integer selected in $\{1, \ldots, \min\{X_h, |\mathscr{W}|\}\}$, where X_h denotes the *randomization factor* of each heuristic $h \in \mathscr{H}$. A small value for X_h implies the tours constructed will be similar to the tour provided by the deterministic version of the heuristic h ($X_h = 1$ is exactly the deterministic heuristic), whereas a large value for X_h allows a greater diversification of the tours. Assuming that $\mathscr{W}$ is updated every time a customer is inserted into p_0^k, the four sampling heuristics operate as follows:

- RNN: Set $p_0^k = (0)$ and vertex $v = 0$. At each iteration of RNN: Sort $\mathscr{W}$ in nondecreasing order of $t_{(v,w)}$. Append $w = \mathscr{W}_{(x)}$ (i.e., the xth element in the ordered set $\mathscr{W}$) to p_0^k and set $v = w$. Stop when $|\mathscr{W}| = 0$ and then append 0 to p_0^k to complete a tour.
- RNI: Initialize p_0^k as a tour starting at the depot and performing a round trip to a randomly selected customer (henceforth this procedure will be referred to as initialize p_0^k). At each iteration of RNI: Sort $\mathscr{W}$ in nondecreasing order of $t_{\min}(w)$. Insert $w = \mathscr{W}_{(x)}$ in p_0^k in the position with the smallest increment in the cost of the tour. Stop when $|\mathscr{W}| = 0$.
- RFI: Initialize p_0^k. At each iteration of RFI: Sort $\mathscr{W}$ in nondecreasing order of $t_{\max}(w)$ and insert $w = \mathscr{W}_{(x)}$ in the best possible position in the tour p_0^k. Stop when $|\mathscr{W}| = 0$.
- RBI: Initialize p_0^k. At each iteration of RBI: Sort $\mathscr{W}$ in nondecreasing order of $\Lambda_{\min}(w)$ and insert $w = \mathscr{W}_{(x)}$ in the best possible position in the tour p_0^k. Stop when $|\mathscr{W}| = 0$.

3.3. Splitting Procedure

The giant tour $p^k = (0, v_{(1)}, \ldots, v_{(i)}, \ldots, v_{(n)}, 0)$ built at iteration k is split into feasible routes by means of an auxiliary acyclic and directed graph $G' = (\mathscr{V}', \mathscr{A}')$, where $\mathscr{V}' = \{0, \ldots, n\}$ is the vertex set and $\mathscr{A}'$ is the arc set. Set $\mathscr{A}'$ contains arc (i, j), $i < j$, if route $r_{(i,j)} = (0, v_{(i+1)}, v_{(i+2)}, \ldots, v_{(j)}, 0)$ does not violate the vehicle capacity and satisfies all duration-related constraints in $\mathscr{C}$. The weight of each arc $(i, j) \in \mathscr{A}'$ is the expected duration of route $r_{(i,j)}$, $E[\tilde{T}_{r_{(i,j)}}]$. The verification that the route satisfies the constraints in $\mathscr{C}$ is done by invoking $\mathtt{PH}(r_{(i,j)})$. An optimal partition of p^k into a solution corresponds to a shortest path from 0 to n in G'. The splitting procedure simultaneously builds the arc set (corresponding to all feasible routes) $\mathscr{A}'$ and finds a shortest path. The reader is referred to

Prins, Lacomme, and Prodhon (2014) for an in-depth survey of splitting procedures.

Figure 9 illustrates how the splitting procedure works. The auxiliary graph is built at each iteration k based on p^k such that each arc represents a feasible route. For instance, in iteration $k=1$, arc $(1,3)$ of the auxiliary graph represents route $(0,2,3,0)$. The shortest path from 0 to 5 in the auxiliary graph, highlighted with bold arcs, is mapped to its associated solution s^k. Also, the set of arcs in the auxiliary graph is mapped to the set of feasible routes Ω^k.

Algorithm 2 (Splitting procedure)
Input: G, VRP graph; Q, vehicle capacity; $\mathscr{C}$, set of constraints; p^k, giant TSP-like tour.
Output: $\langle \Omega^k, s^k\rangle$, set of feasible routes and best solution given the order of customers in p^k.

1: $c_0 \leftarrow 0$
2: **for** $i=1$ **to** n **do**
3: $\quad c_i \leftarrow \infty$
4: **end for**
5: $\Omega^k \leftarrow \varnothing$
6: **for** $i=0$ **to** n **do**
7: $\quad$ `load` $\leftarrow 0$
8: $\quad j \leftarrow i+1$
9: $\quad$ **repeat**
10: $\quad\quad r \leftarrow r_{(i,j)}$
11: $\quad\quad$ `load` $\leftarrow$ `load` $+ d_{v_{(j)}}$
12: $\quad\quad$ `continue` $\leftarrow$ `false`
13: $\quad\quad$ **if** `load` $\leq Q$, **then**
14: $\quad\quad\quad \mathrm{PH}_r \leftarrow$ Call `PH`(r)
15: $\quad\quad\quad$ **if** `isRouteFeasible`(PH_r), **then**
16: $\quad\quad\quad\quad \Omega^k \leftarrow \Omega^k \cup \{r\}$
17: $\quad\quad\quad\quad E[\tilde{T}_r] \leftarrow$ `getExpectedValue`(PH_r)
18: $\quad\quad\quad\quad$ **if** $c_{i-1} + E[\tilde{T}_r] \leq c_j$, **then**
19: $\quad\quad\quad\quad\quad c_j \leftarrow c_{i-1} + E[\tilde{T}_r]$
20: $\quad\quad\quad\quad\quad P_j \leftarrow i-1$
21: $\quad\quad\quad\quad$ **end if**
22: $\quad\quad\quad\quad$ `continue` $\leftarrow$ `true`
23: $\quad\quad\quad$ **else if** `isRouteExtensible`(PH_r), **then**
24: $\quad\quad\quad\quad$ `continue` $\leftarrow$ `true`
25: $\quad\quad\quad$ **end if**
26: $\quad\quad$ **end if**
27: $\quad\quad j \leftarrow j+1$
28: $\quad$ **until** $j > n$ **or** `continue` $=$ `false`
29: **end for**
30: $s^k \leftarrow$ `retrieveSolution`(p^k, P)
31: **return** $\langle \Omega^k, s^k\rangle$

Algorithm 2 describes the splitting procedure. In lines 1–4 the algorithm initializes the label c_i for the cost of the shortest path from 0 to each vertex $i \in \mathscr{V}'$. In line 5 the algorithm initializes the set Ω^k of sampled routes for iteration k. Then the procedure implicitly evaluates every route r that is a candidate to join Ω^k. The latter is done by iterating over the arcs' tails (iterator i from lines 6 to 29) and heads (iterator j from lines 9 to 28). At each pass of the inner loop (lines 9 to 28), the algorithm considers a new route r by appending an additional customer to the last evaluated route (line 10). We refer to this operation as *route extension*. Next the procedure evaluates the capacity constraint of route r (line 13). If the route is feasible in terms of the capacity, the algorithm calls `PH`(r) in line 14 to retrieve PH_r with the necessary performance metrics. Then the algorithm makes a call to the Boolean function `isRouteFeasible`(PH_r) to check whether the route satisfies the constraints in $\mathscr{C}$ (line 15). If route r is feasible, the algorithm adds the route to Ω^k (line 16) and the procedure updates the shortest path labels in lines 18–21. Note that even if route r turns out to be infeasible, extending the route may lead to a feasible route r' if constraints in $\mathscr{C}$ are nonmonotonic. The algorithm makes a call to the Boolean function `isRouteExtensible`(PH_r) (line 23) to check if route r is *extensible* and, depending on the result, the algorithm decides to break (or not) the inner loop. Online Appendix C illustrates the concept of nonmonotonic constraints and route extensibility with a specific example. Finally, the algorithm calls the function `retrieveSolution`(p^k, P), which takes as input tour p^k and the array of predecessors P and maps the shortest path in G' to its corresponding solution (set of routes) in G (line 30).

Note that some routes may be sampled several times during the algorithm's execution. For instance, the fundamental routes $(0, v_i, 0), v_i \in \mathscr{V}$, are sampled in every MSH iteration. To avoid unnecessary computations, our `PH`(r) evaluator has a dictionary of previously evaluated routes. In other words, when `PH`(r) is called over a previously evaluated route (line 14), it directly retrieves the associated PH_r without recomputing the convolutions of travel and service times.

3.4. Set Partitioning

The mapping phase builds the set $\mathscr{R}$ of routes belonging to the final solution by solving a set partitioning formulation over the set Ω of sampled routes using s^* as an initial feasible solution and $f(s^*)$ as an upper bound. The problem can be formulated as follows:

$$\min_{\mathscr{R}\subseteq\Omega}\left\{\sum_{r\in\mathscr{R}} E[\tilde{T}_r]\colon \bigcup_{r\in\mathscr{R}} r=\mathscr{V};\, r_i\cap r_j=\{0\},\, \forall\, r_i, r_j\in\mathscr{R}\right\}. \tag{2}$$

The satisfaction of constraints in $\mathscr{C}$ is assured in the sampling phase, as all of the routes that join Ω during the splitting procedure are feasible. The set partitioning ensures that each customer is visited exactly once. Note that, if the fundamental route $(0, v_i, 0)$ is feasible for every customer v_i, then the set partitioning problem over Ω is always feasible. The task of solving this problem is left to a commercial mixed-integer linear optimizer.

4. Solving the Distance-Constrained Capacitated VRP with Stochastic Travel and Service Times

To illustrate the proposed approach, we present an application to the distance-constrained capacitated vehicle routing problem with stochastic travel and service times (DCVRPSTT). The DCVRPSTT, introduced by Laporte, Louveaux, and Mercure (1992), consists of designing a route set $\mathcal{R}$ of minimal total expected duration whereas ensuring a certain *service level*. The following service level condition enforces that every route $r \in \mathcal{R}$ finishes before threshold T with a probability greater than β:

$$P(\tilde{T}_r \leq T) \geq \beta \quad \forall\, r \in \mathcal{R}. \tag{3}$$

In other words, in this problem variant, set $\mathcal{C}$ includes only the set of constraints (3). The two performance measures of interest in the DCVRPSTT are $E[\tilde{T}_r]$ and $P(\tilde{T}_r \leq T)$.

4.1. Test Instances

To conduct our experiments we adapted instances from the distance-constrained capacitated vehicle routing problem (DCVRP) by Christofides, Mingozzi, and Toth (1979) to the DCVRPSTT. We chose three instances with nonzero service times and a number of customers ranging between 50 and 100. To account for the stochastic components (travel and service times) we assigned a probability distribution, whose expected value matches the values reported in the original instance.

For each instance we constructed three scenarios, assuming in each one a probability distribution for the travel times with different characteristics. In the first scenario for all travel times we assume an Erlang distribution, which has low variance. In the second scenario, we assume that all travel times follow a lognormal distribution, which has a comparatively high variance. In the third scenario, we assume that the underlying distribution for travel times is based on a Burr distribution, a heavily skewed distribution with a long tail to the right. In all scenarios we set the service level $\beta = 0.85$ and set the time limit threshold T to the value of the original distance constraint reported in the instances.

4.2. Travel and Service Time Distributions

In this section we describe the distributions assumed for each scenario. In all cases we describe a base distribution $\mathcal{X}$. The distribution for travel time $\tilde{t}_{ij}$ is of the form $a\mathcal{X}$, $a > 0$, so that $E[a\mathcal{X}]$ is equal to the original travel time between customers i and j.

4.2.1. Scenario 1: Erlang Distribution. In the first scenario we consider an Erlang distribution with four phases. This distribution has a relatively low squared coefficient of variation[2] of 0.25 and a low skewness[3] of 0.5. Such a distribution may be adequate when travel times have low variability, as is the case in uninterrupted traffic at free flow speed.

4.2.2. Scenario 2: Lognormal Distribution. Lognormal distributions have often been used to model travel times of congested links in uninterrupted traffic. The main difficulty when dealing with lognormal distributions is that there is no exact method for computing the convolutions. Although simulation is the most common way to circumvent this difficulty, we advocate for an alternative approach: approximating the lognormal distributions with PH distributions and computing the convolutions using the formulae presented in Online Appendix B. For this scenario, $\mathcal{X}$ follows a lognormal $(0, 1)$ distribution, which has a relatively high squared coefficient of variation equal to 1.7 and a high skewness of 6.2.

4.2.3. Scenario 3: Burr Distribution. Travel times in interrupted traffic (e.g., urban corridors) under heavy congestion may have very high variance and skewness, representing a long tail to the right of the distribution. Under these conditions, Susilawati, Taylor, and Somenahalli (2013) found that Burr Type XII distributions fit travel times better than lognormal, gamma, Weibull, and normal distributions. Therefore, in this scenario we use a Burr distribution with cumulative distribution function $1 - (1 + x^{\gamma_1})^{-\gamma_2}$ with $\gamma_1 = 2$ and $\gamma_2 = 1$. This choice of parameters leads to a heavy tail, since it makes the second and higher moments of this distribution infinite.

4.2.4. Service Times. In each scenario we consider two possible distributions for the service times: they are either deterministic or they follow an exponential distribution. In both cases the expected value of the service times is equal to the service time reported in the original instance.

4.3. Route Evaluators

We describe how to model travel and service times using a *normal route evaluator*, which fits all random times to the family of normal distributions and exploits the properties of such distributions to efficiently compute the required convolutions and probabilities. Next, we describe a *simulation route evaluator*, which generates random variates to estimate the distribution function of a route duration. Finally, we describe the suggested *PH route evaluator*, which allows us to easily fit the distributions from empirical data and compute the convolutions efficiently.

[2] The exponential distribution with a squared coefficient of variation of 1 is considered neutral; a value of 0.25 is considered as a low squared coefficient of variation.

[3] A skewness of 0, like the normal distribution, implies a symmetric distribution. A high positive skewness indicates a longer tail to the right of the distribution.

4.3.1. Normal Route Evaluator—NOR(r). To approximate a travel or service time with mean μ and variance σ^2, one can use a normal distribution $\mathcal{N}(\mu, \sigma^2)$. The convolution of n_r normal random variables is normally distributed as $\mathcal{N}(\sum_{i=1}^{n_r}\mu_i, \sum_{i=1}^{n_r}\sigma_i^2)$. This approximation fits the first two moments of the distribution exactly but disregards all higher moments.

In the case of the Burr distribution, however, we cannot use the same approach, since σ^2 is infinite for our choice of parameters. To overcome this difficulty, we generate random variates from the Burr distribution and use the sample mean and variance as parameters for the normal distribution.

4.3.2. Simulation Route Evaluator—SIM(r). We can estimate $P(\tilde{T}_r \le T)$ using Algorithm 3. The methodology for generating random variates from the distribution of travel and service times (line 8) depends on the specific distribution. We use SSJ (acronym for Stochastic Simulation in Java) by L'Ecuyer (2012) to generate the exponential, Erlang, and lognormal random variates. For the Burr distribution, we use numerical inversion of the distribution function. In all cases, we use $N_0 = 1{,}000$ replications, as suggested by Li, Tian, and Leung (2010).

Algorithm 3 (Stochastic simulation)

Input: r, route to evaluate; $\{\tilde{t}_e\}_{e\in r}$, random variables describing travel times for edges $e \in r$; $\{\tilde{s}_v\}_{v\in r}$, random variables describing service times for customers $v \in r$; T, threshold; β, required probability; N_0, number of random variates to generate.

Output: `true` if $P(\tilde{T}_r \le T) \ge \beta$, `false` otherwise.

1: $j \leftarrow 0$ ▹ j represents number of replications in which $\tilde{T}_r \le T$
2: **for** $i = 1$ **to** N_0 **do**
3: $\quad\tilde{T}_r^i \leftarrow 0$ ▹ $\tilde{T}_r^i$ represents the total duration in one replication
4: $\quad$**for all** $e \in r$ **do**
5: $\quad\quad\tilde{T}_r^i \leftarrow \tilde{T}_r^i + \texttt{generate}(\tilde{t}_e)$ ▹ Generates one sample from distribution $\tilde{t}_e$
6: $\quad$**end for**
7: $\quad$**for all** $v \in r$ **do**
8: $\quad\quad\tilde{T}_r^i \leftarrow \tilde{T}_r^i + \texttt{generate}(\tilde{s}_v)$ ▹ Generates one sample from distribution $\tilde{s}_v$
9: $\quad$**end for**
10: $\quad$**if** $\tilde{T}_r^i \le T$, **then**
11: $\quad\quad j \leftarrow j + 1$
12: $\quad$**end if**
13: **end for**
14: **if** $j/N_0 \ge \beta$, **then**
15: $\quad$**return** `true`
16: **else**
17: $\quad$**return** `false`
18: **end if**

4.3.3. PH Route Evaluator—PH(r). The exponential distribution used to model a service time with mean $1/\alpha$ is a special case of PH distribution $PH(\boldsymbol{\tau}_\mathbf{S}, \mathbf{Z}_\mathbf{S})$ with one phase, where

$$\boldsymbol{\tau}_\mathbf{S} = (1),$$
$$\mathbf{Z}_\mathbf{S} = (-\alpha).$$

Similarly, the Erlang distribution with four phases and rate λ used in scenario 1 is a special case of PH distribution $PH(\boldsymbol{\tau}_\mathrm{E}, \mathbf{Z}_\mathrm{E})$ with four phases, where

$$\boldsymbol{\tau}_\mathrm{E} = (1 \quad 0 \quad 0 \quad 0),$$
$$\mathbf{Z}_\mathrm{E} = \begin{pmatrix} -\lambda & \lambda & 0 & 0 \\ 0 & -\lambda & \lambda & 0 \\ 0 & 0 & -\lambda & \lambda \\ 0 & 0 & 0 & -\lambda \end{pmatrix}.$$

To fit the other distributions we generate random variates from each distribution and fit the PH distribution using the algorithm proposed by Thummler, Buchholz, and Telek (2006), as illustrated in §2.2. In the case of the lognormal(0, 1) distribution, the fitted distribution is a $PH(\boldsymbol{\tau}_\mathrm{L}, \mathbf{Z}_\mathrm{L})$ with four phases, where

$$\boldsymbol{\tau}_\mathrm{L} = (0.68 \quad 0.00 \quad 0.31 \quad 0.01),$$
$$\mathbf{Z}_\mathrm{L} = \begin{pmatrix} -2.09 & 2.09 & 0.00 & 0.00 \\ 0.00 & -2.09 & 0.00 & 0.00 \\ 0.00 & 0.00 & -0.33 & 0.00 \\ 0.00 & 0.00 & 0.00 & -0.07 \end{pmatrix}.$$

For the Burr distribution the fitting algorithm yields a $PH(\boldsymbol{\tau}_\mathrm{B}, \mathbf{Z}_\mathrm{B})$ distribution with nine phases, where

$$\boldsymbol{\tau}_\mathrm{B} = (0.17\ 0.00\ 0.00\ 0.51\ 0.00\ 0.26\ 0.05\ 0.00\ 0.01),$$
$$\mathbf{Z}_\mathrm{B} = \begin{pmatrix} -6.47 & 6.47 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & -6.47 & 6.47 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & -6.47 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & -1.71 & 1.71 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & -1.71 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & -0.42 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & -0.92 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & -1.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & -0.05 \end{pmatrix}.$$

4.4. Computational Results

We compare the performances of the state-of-the-art routing algorithm (MSH) coupled with the three different *route evaluators*. We refer to the algorithm running with the normal, simulation, and PH route evaluator as MSH(NOR), MSH(SIM), and MSH(PH), respectively. Every experiment was repeated 20 times with $K = 1{,}000$ iterations during the sampling phase of MSH, and we report the average performance. To solve the underlying set partitioning in MSH we use

Table 1 Solutions for Scenario 1: Erlang Travel Times Using the Different Route Evaluators

Service distribution	Instance	Route evaluator	Expected duration: Travel	Expected duration: Service	Routes: Total	Routes: Infeasible	Routes: % Inf.	Routes: Viol. (%)	Total time (s)	Comparison: Δ_{EV} (%)	Comparison: Δ_{Inf} (%)
Deterministic	D-051-06c	NOR(r)	597.9	500	7.0	0.0	0.0	0.0	25	0.0	0.0
		SIM(r)	597.8	500	7.0	0.1	1.4	0.1	21	0.0	1.4
		PH(r)	597.9	500	7.0	0.0	0.0	0.0	46	—	—
	D-076-11c	NOR(r)	1,048.9	750	13.9	0.0	0.0	0.0	33	0.1	0.0
		SIM(r)	1,042.2	750	13.8	2.1	15.2	0.7	28	−0.6	15.2
		PH(r)	1,048.0	750	13.8	0.0	0.0	0.0	46	—	—
	D-101-09c	NOR(r)	933.5	1,000	10.0	0.0	0.0	0.0	262	0.0	0.0
		SIM(r)	932.7	1,000	10.0	0.3	2.9	0.2	292	−0.1	2.9
		PH(r)	933.4	1,000	10.0	0.0	0.0	0.0	489	—	—
Exponential	D-051-06c	NOR(r)	619.8	500	7.0	0.0	0.0	0.0	23	0.0	0.0
		SIM(r)	617.3	500	7.0	0.6	8.6	0.6	23	−0.4	8.6
		PH(r)	619.8	500	7.0	0.0	0.0	0.0	52	—	—
	D-076-11c	NOR(r)	1,145.3	750	15.9	0.0	0.0	0.0	32	0.0	0.0
		SIM(r)	1,126.6	750	15.2	3.0	19.7	2.4	36	−1.6	19.7
		PH(r)	1,144.9	750	15.9	0.0	0.0	0.0	50	—	—
	D-101-09c	NOR(r)	980.1	1,000	11.0	0.0	0.0	0.0	150	0.0	0.0
		SIM(r)	985.4	1,000	11.0	0.2	1.8	0.2	33	0.5	1.8
		PH(r)	980.1	1,000	11.0	0.0	0.0	0.0	369	—	—

Gurobi 5.5.0 (Gurobi Optimization 2014). Finally, to verify the solution quality, we check the probability that the routes complete before the threshold T using an a posteriori stochastic simulation with $N_0 = 1{,}000{,}000$.

Tables 1–3 present the results for the different underlying distributions. From left to right, the tables show, for each of the service time distributions and route evaluators, the expected travel and service time; the average number of routes, the average number of infeasible routes, the percentage of infeasible routes in the solution (given by the a posteriori stochastic simulation), and the average violation of the worst route given by $\max\{0, \max_{r\in\mathcal{R}}\{\beta - P(\tilde{T}_r \leq T)\}\}$; the total computational time; and the comparison of each algorithm configuration against MSH(PH) in terms of solution expected value and percentage of infeasible routes, computed as

$$\Delta_{EV} = \frac{\texttt{MSH}(\cdot)\text{travel time} - \texttt{MSH(PH)}\text{travel time}}{\texttt{MSH(PH)}\text{travel time}} \times 100\%,$$

$$\Delta_{Inf} = \texttt{MSH}(\cdot)\%\text{ infeasible} - \texttt{MSH(PH)}\%\text{ infeasible}.$$

Table 2 Solutions for Scenario 2: Lognormal Travel Times Using the Different Route Evaluators

Service distribution	Instance	Route evaluator	Expected duration: Travel	Expected duration: Service	Routes: Total	Routes: Infeasible	Routes: % Inf.	Routes: Viol. (%)	Total time (s)	Comparison: Δ_{EV} (%)	Comparison: Δ_{Inf} (%)
Deterministic	D-051-06c	NOR(r)	772.0	500	10.0	0.0	0.0	0.0	15	18.3	0.0
		SIM(r)	621.9	500	7.1	1.8	24.8	2.3	15	−4.8	24.8
		PH(r)	652.4	500	8.0	0.0	0.0	0.0	31	—	—
	D-076-11c	NOR(r)[a]									
		SIM(r)	1,234.9	750	17.0	7.6	45.2	0.9	25	−8.5	45.2
		PH(r)	1,350.2	750	18.9	0.0	0.0	0.0	40	—	—
	D-101-09c	NOR(r)	1,110.4	1,000	13.0	0.0	0.0	0.0	111	10.2	0.0
		SIM(r)	989.5	1,000	11.0	1.3	11.7	2.1	169	−1.9	11.7
		PH(r)	1,007.3	1,000	11.3	0.0	0.0	0.0	338	—	—
Exponential	D-051-06c	NOR(r)	800.9	500	10.4	0.0	0.0	0.0	16	20.5	0.0
		SIM(r)	654.3	500	8.0	1.4	17.5	1.3	14	−1.6	17.5
		PH(r)	664.9	500	8.3	0.0	0.0	0.0	38	—	—
	D-076-11c	NOR(r)[a]									
		SIM(r)	1,310.2	750	18.0	7.8	43.3	1.9	22	−9.5	43.3
		PH(r)	1,448.4	750	20.0	0.0	0.0	0.0	43	—	—
	D-101-09c	NOR(r)	1,164.3	1,000	14.0	0.0	0.0	0.0	160	11.9	0.0
		SIM(r)	1,030.6	1,000	12.0	1.9	15.8	1.0	128	−0.9	15.8
		PH(r)	1,040.6	1,000	12.0	0.0	0.0	0.0	286	—	—

[a]The route evaluator is unable to find any feasible solution.

Note that to compute Δ_{EV}, we use only the travel times (instead of the total duration), since the expected service time is the same regardless of the route evaluator used or the service time distribution assumed (provided that the different service time distributions have the same expected value, as is the case in this paper). Indeed, each customer is visited exactly once in all cases, and the expected service duration per customer is the same. For instance, if there are 50 customers and each requires a service time whose expected value is 10, then the expected value of the total service time spent at the customers sites is 500, independent of the number of vehicles used or the sequence of visits.

4.4.1. Results for Scenario 1: Erlang. As Table 1 shows, when the underlying distribution for travel times is Erlang, MSH(NOR) and MSH(PH) deliver similar high-quality solutions in all instances. Algorithm MSH(SIM) provides solutions with shorter total expected travel times. Nonetheless, these solutions tend to contain routes that are found a posteriori to be infeasible. In particular, in instance D-076-11c with exponential service times, the solution delivered by MSH(SIM) has an improvement of 1.6% in the expected travel time at the expense of having 19.7% infeasible routes in the solution, with an average worst route violation of 2.4%. In terms of computational times, MSH(NOR) and MSH(SIM) report similar execution times. In contrast, MSH(PH) reports times that are between 20 and 240 seconds longer, depending on the instance.

In conclusion, for this family of distributions, NOR(r) seems to be the best choice, since it achieves good accuracy with small computational effort. Indeed, since this particular Erlang distribution's skewness is close to 0, a normal distribution, which has a skewness of 0, could be a good fit in terms of the third moment of the distribution. In this case PH(r) provides perfect accuracy (since both the exponential and Erlang distributions are special cases of PH distributions) in exchange for a computational overhead. On the other hand, SIM(r) is arguably the worst route evaluator in this case, delivering less accuracy with the same computational effort as NOR(r).

4.4.2. Results for Scenario 2: Lognormal. Normal distributions cannot accurately capture the characteristics of the family of lognormal distributions chosen for the travel times, such as the high skewness, and are therefore a poor fit. As Table 2 shows, NOR(r) is overly pessimistic (i.e., overestimates the probability of violating the duration constraint), and in consequence MSH(NOR) delivers solutions with more (shorter) routes than MSH(SIM) and MSH(PH), which translates into solutions with 8%–20% longer total expected travel times. Moreover, NOR(r) is so pessimistic on instance D-076-11c (with both deterministic and exponential service times) that the algorithm is unable to find a feasible solution, although such a solution exists.

Simulation relies on a discrete distribution with finite support to approximate the travel and service times. Heavy-tailed distributions, such as the lognormal, are hard to approximate using this approach because tail probabilities may be underrepresented. As Table 2 shows, MSH(SIM) is overly optimistic, leading in some cases to solutions in which half the routes turn out to be infeasible, with an average worst route violation of up to 2.3% when evaluated a posteriori.

Algorithm MSH(PH), in contrast, reports feasible solutions whose expected total travel times are only slightly longer than the (infeasible) solutions found using MSH(SIM) and significantly shorter than the solutions delivered by MSH(NOR). In exchange, MSH(PH) incurs in a computational overhead, compared to the other algorithm configurations.

4.4.3. Results for Scenario 3: Burr. All three route evaluators are challenged by the heavy-tail and infinite variance of the Burr distribution selected for the travel times (Table 3). The normal distribution is the worst fit; NOR(r) did not find any feasible solution. In instance D-051-06c, both MSH(SIM) and MSH(PH) report similar solutions, with average worst route violations of less than 0.7% and 0.6%, respectively. However, in instances D-076-11c and D-101-09c, PH(r) provides the most accurate estimation of the actual route duration, leading to solutions with fewer infeasible routes, lower average violation, and little penalty in terms of expected travel time. Even though it is not a perfect fit, MSH(PH) is able to achieve very low average worst route violations of less than 1.2% across all instances.

For the Burr distribution, the computational time for MSH(PH) increases compared to the other algorithm configurations. Indeed, for this case, the PH distribution has nine phases instead of four, leading to longer computational times.

4.4.4. Service Time Considerations. In all instances, the introduction of additional variability in the form of exponential service times (compared to deterministic service times) results in shorter routes to ensure satisfaction of the chance constraints (3). The solutions are then composed of a larger number of (shorter) routes and longer expected total travel times. Moreover, assuming exponential service times results in fewer feasible routes found during the sampling phase of MSH, which leads to fewer variables in the set partitioning problem and faster solution times (especially in instance D-101-09c).

4.4.5. Closing Remarks. We found that assuming normal distributions is a good alternative when the underlying travel times do not exhibit large skewness. However, in routing problems, more often than

Table 3 **Solutions for Scenario 3: Burr Travel Times Using the Different Route Evaluators**

Service distribution	Instance	Route evaluator	Expected duration: Travel	Expected duration: Service	Routes: Total	Routes: Infeasible	Routes: % Inf.	Routes: Viol. (%)	Total time (s)	Comparison: Δ_{EV} (%)	Comparison: Δ_{Inf} (%)
Deterministic	D-051-06c	NOR(r)[a]									
		SIM(r)	618.8	500	7.3	1.6	21.1	0.7	16	−0.1	0.0
		PH(r)	619.3	500	7.3	1.6	21.1	0.6	204	—	—
	D-076-11c	NOR(r)[a]									
		SIM(r)	1,164.9	750	15.9	6.2	39.2	2.3	31	−3.1	15.6
		PH(r)	1,201.8	750	16.5	3.8	23.6	1.2	150	—	—
	D-101-09c	NOR(r)[a]									
		SIM(r)	974.2	1,000	11.0	2.6	23.4	1.8	216	−1.4	10.2
		PH(r)	988.5	1,000	11.4	1.5	13.2	1.0	1,821	—	—
Exponential	D-051-06c	NOR(r)[a]									
		SIM(r)	654.1	500	8.0	0.2	2.5	0.1	13	0.0	1.2
		PH(r)	654.2	500	8.0	0.1	1.3	0.1	199	—	—
	D-076-11c	NOR(r)[a]									
		SIM(r)	1,259.7	750	17.2	7.3	42.4	2.4	25	−1.9	19.9
		PH(r)	1,284.4	750	18.0	4.1	22.5	0.4	156	—	—
	D-101-09c	NOR(r)[a]									
		SIM(r)	1,021.5	1,000	11.9	3.4	28.6	1.6	15	−0.4	15.3
		PH(r)	1,025.4	1,000	12.0	1.6	13.3	0.6	1,580	—	—

[a]The route evaluator is unable to find any feasible solution.

not, travel times have a positive skewness and normal distributions that lead to poor routing decisions. Routing algorithms must therefore consider skewness as an important factor to lead to reliable routing decisions. In contrast, using Monte Carlo simulation to evaluate routes seems to lead in all cases to overly optimistic solutions that do not satisfy the chance constraints (3) in practice. If the routing algorithm uses PH distributions to evaluate routes instead, it gets solutions with similar total expected durations that are much more reliable in practice. Phase type distributions are the most accurate choice when modeling travel and service times for all distributions used in our experiments.

5. Conclusions

This paper proposes the use of PH distributions to accurately model the distributions of travel and service times and to compute their convolutions in an exact and algorithmically tractable manner. Several key performance metrics can then be computed from the distributions. By embedding them in a routing engine, PH distributions allow us to tackle VRPSTT without the limitation of working with a reduced set of known distributions or using approximations to compute the convolutions.

We conducted computational experiments with different route evaluators previously used in the VRPSTT literature. We observed that focusing solely on expected values and variances (e.g., normal distributions), whereas neglecting other stochastic aspects of the problem, may lead to expensive routing decisions, depending on the underlying distributions present in the routing problem. Furthermore, using simulation may lead to underrepresentation of the tail probabilities, leading to unreliable solutions. Moreover, we showed that modeling stochastic times with PH distributions allows us to easily incorporate a given service level, leading to reliable solutions, regardless of the distributions of the travel and service times. Finally, the quality of the solutions found depends on the travel and service time distributions, showing the importance of accurately modeling these times.

Although we limited our experiments to cases in which all travel times follow the same family of distributions, this is by no means a limitation of the proposed approach. In fact, our approach is suitable to real-life scenarios, as long as there is sufficient empirical data. Indeed, one could independently fit the travel time distribution of each edge in $\mathcal{E}$ and the service time distribution of each vertex in $\mathcal{V}$. The PH route evaluator would then take into account, during the solution process, the characteristics of all travel and service times without making unrealistic assumptions.

We focused only on the DCVRPSTT, but PH distributions could be used to tackle other stochastic VRP variants. One such variant is the VRPSTT with time windows, where each customer should be serviced between a specified earliest and latest time. Because the arrival time at each customer is a random variable, it cannot be guaranteed that the customer will be serviced in his time window. However, one could impose the constraint that the probability of visiting a customer within the specified time window should meet a predefined service level. Computing the arrival time distribution at each customer can be done efficiently using PH distributions.

Another variant that could be tackled is the VRP with stochastic demands, in which the quantity requested by each customer is not known a priori, but follows a probabilistic distribution. Because each vehicle has a limited capacity, a route failure may occur and a recourse action must be undertaken (usually a return trip to the depot). The goal is to minimize the expected route duration after recourse. The computation of the probability that a route fails when servicing a specific customer can be done efficiently if the demand distributions have the additive property. However, by modeling these distributions with PH distributions, the additive property is no longer necessary and the problem could be solved under more general conditions.

Supplemental Material

Supplemental material to this paper is available at http://dx.doi.org/10.1287/trsc.2015.0601.

Acknowledgments

The authors thank the editor-in-chief, associate editor, and anonymous reviewers for their constructive comments that improved the paper. As always, any and all errors are fully attributable to the authors.

References

Bobbio A, Horvát A, Telek M (2005) Matching three moments with minimal acyclic phase type distributions. *Stochastic Models* 21(2–3):303–326.

Boyles SD, Voruganti A, Waller ST (2010) *Quantifying Travel Time Variability in Transportation Networks* (Southwest Region University Transportation Center, College Station, TX).

Chen L, Hà MH, Langevin A, Gendreau M (2014) Optimizing road network daily maintenance operations with stochastic service and travel times. *Transportation Res. Part E: Logist. Transportation Rev.* 64:88–102.

Christofides N, Mingozzi A, Toth P (1979) The vehicle routing problem. Christofides N, Mingozzi A, Toth P, eds. *Combinatorial Optimization* (Wiley, Chichester, UK), 315–338.

Doerner K, Schmid V (2010) Survey: Matheuristics for rich vehicle routing problems. Talbi E, ed. *Hybrid Metaheuristics*, Lecture Notes Comput. Sci., Vol. 6373 (Springer, Berlin Heidelberg), 206–221.

Errico F, Desaulniers A, Gendreau M, Rei W, Rousseau LM (2013) The vehicle routing problem with hard time windows and stochastic service times. Technical report G 2013 45, GERAD, Montréal. https://www.gerad.ca/en/papers/G-2013-45.

Figliozzi MA (2010) The impacts of congestion on commercial vehicle tour characteristics and costs. *Transportation Res. Part E: Logist. Transportation Rev.* 46(4):496–506.

Gendreau M, Jabali O, Rei W (2014) Stochastic vehicle routing problems. Toth P, Vigo D, eds. *Vehicle Routing: Problems, Methods, and Applications*, Second ed. (SIAM, Philadelphia), 213–240.

Gendreau M, Laporte G, Potvin J-Y (2001) Metaheuristics for the capacitated VRP. Toth P, Vigo D, eds. *The Vehicle Routing Problem* (SIAM, Philadelphia), 129–154.

Gurobi Optimization Inc. (2014) Gurobi optimizer reference manual. Accessed June 24, 2015, http://www.gurobi.com.

Kenyon AS, Morton DP (2003) Stochastic vehicle routing with random travel times. *Transportation Sci.* 37(1):69–82.

Laporte G (2007) What you should know about the vehicle routing problem. *Naval Res. Logist.* 54(8):811–819.

Laporte G, Louveaux F, Mercure H (1992) The vehicle routing problem with stochastic travel times. *Transportation Sci.* 26(3): 161–170.

Latouche G, Ramaswami V (1999) *Introduction to Matrix Analytic Methods in Stochastic Modeling* (SIAM, Philadelphia).

Lecluyse C, Van Woensel T, Peremans H (2009) Vehicle routing with stochastic time-dependent travel times. *4OR* 7(4):363–377.

L'Ecuyer P (2012) SSJ: Stochastic simulation in Java. Accessed June 24, 2015, http://www.iro.umontreal.ca/~simardr/ssj/indexe.html.

Li X, Tian P, Leung SCH (2010) Vehicle routing problems with time windows and stochastic travel and service times: Models and algorithm. *Internat. J. Production Econom.* 125(1):137–145.

Mendoza JE, Villegas JG (2013) A multi-space sampling heuristic for the vehicle routing problem with stochastic demands. *Optim. Lett.* 7(7):1503–1516.

Neuts MF (1981) *Matrix-Geometric Solutions in Stochastic Models: An Algorithmic Approach* (Dover Publications Inc., New York).

Prins C (2004) A simple and effective evolutionary algorithm for the vehicle routing problem. *Comput. Oper. Res.* 31(1):1985–2002.

Prins C, Lacomme P, Prodhon C (2014) Order-first split-second methods for vehicle routing problems: A review. *Transportation Res. Part C: Emerging Technologies* 40:179–200.

Susilawati S, Taylor M, Somenahalli S (2013) Distributions of travel time variability on urban roads. *J. Advanced Transportation* 47(8):720–736.

Taş D, Dellaert N, Van Woensel T, de Kok T (2013) Vehicle routing problem with stochastic travel times including soft time windows and service costs. *Comput. Oper. Res.* 40(1):214–224.

Taş D, Gendreau M, Dellaert N, Van Woensel T, de Kok AG (2014) Vehicle routing with soft time windows and stochastic travel times: A column generation and branch-and-price solution approach. *Eur. J. Oper. Res.* 236(3):789–799.

Telek M, Heindl A (2002) Matching moments for acyclic discrete and continuous phase-type distributions of second order. *Internat. J. Simulation* 3(3–4):47–57.

Thummler A, Buchholz P, Telek M (2006) A novel approach for phase-type fitting with the EM algorithm. *IEEE Trans. Dependable Secure Comput.* 3(3):259–269.

Van Woensel T, Kerbache L, Peremans H, Vandaele N (2003) A vehicle routing problem with stochastic travel times. *Fourth Aegean Internat. Conf. Anal. Manufacturing Systems, Samos, Greece*.

Van Woensel T, Kerbache L, Peremans H, Vandaele N (2007) A queueing framework for routing problems with time-dependent travel times. *J. Math. Modelling Algorithms* 6(1): 151–173.