# Practical Risk Modeling for the Stochastic Technician Routing and Scheduling Problem

Luke Marshall
Microsoft Research, luke.marshall@microsoft.com

Timur Tankayev
Georgia Institute of Technology, timur.tankayev@gatech.edu

Planning for uncertainty is crucial for finding good, stable solutions. However, it is often impractical to incorporate stochastic elements into a large production system. Our paper tackles this issue in the context of the Technician Routing and Scheduling Problem (TRSP). We develop a set of techniques, based on phase-type distributions, to quickly and accurately evaluate risks caused by stochastic service durations. Our framework also supports hard time-windows and time-dependent travel times. We construct a new set of test instances derived from historical data. These instances demonstrate the importance of considering stochasticity and traffic in technician scheduling. We perform an extensive computational analysis over these instances. The experiments show that our approach works well in real-world scenarios and can scale to problem sizes of practical interest.

*Key words*: technician routing and scheduling; stochastic service duration; time-dependent travel; hard time-windows

## 1. Introduction

Many organizations face the problem of scheduling and routing personnel to fulfill service requests [5]. The scope and difficulty of this problem is likely to grow as the demand and competition for field-service increases. Client satisfaction is becoming an important competitive feature, and exploiting new technologies such as predictive analytics and IoT are becoming crucial to address this challenge. There are many variants of the technician routing and scheduling problem (TRSP) [18]. Ours is inspired by the maintenance department within Microsoft, and the associated scheduling tools used internally and by third-party clients. Specifically, we support services with technician-dependent stochastic durations, time-dependent travel, and hard time-windows. The most difficult requirements are associated with hedging against uncertainty in service durations. These service durations are typically unknown before the technician performs an inspection in the field. However, based on the technician's historical performance, it is possible to estimate a probability distribution for the service duration prior to the visit. Planning schedules without taking this uncertainty into account may result in bad routing decisions, missed service time-windows, and employee overtime. The size and scale of real-world problems gives rise to another set of obstacles. Large field-service providers need to schedule thousands of requests over hundreds of technician shifts, often spread over a large geographical area. This leads to travel times between service locations fluctuating significantly due to traffic conditions. In this work we present a practical and scalable solution methodology for TRSP that can deal with all of these issues.

### 1.1. Literature

TRSP can be viewed as a stochastic vehicle routing problem (VRP). [27, 14] and [30] are three excellent (and recent) surveys on the topic. The most efficient solutions to VRP are currently based on set partitioning formulations [28]. In our setting, this corresponds to splitting the services into technician routes. This makes it straightforward to incorporate additional constraints on each route, such as technician skill, tool availability, and maximum travel distances [6, 38]. Given a set partitioning formulation, both the exact [11] and heuristic [16] solution approaches need to evaluate feasible routes. In the stochastic setting, these evaluations need to incorporate risk computations. The *risk* of a route can be used in feasibility constraints [19], as an objective [36], or to inform recourse policies [10, 33]. Sophisticated recourse policies are outside the scope of this paper.

Although stochastic VRP literature is deep and extensive, the existing solution methods do not meet all our requirements. There are a lot of successful approaches for risk hedging based on either sampling representative scenarios [15] or robust formulations [22]. However, such approaches require solving a very large deterministic version of the problem, which proved infeasible for our instance sizes. Alternative approaches are based on explicitly evaluating risk measures for the generated routes. These involve fitting distributions to service durations and carrying out risk calculations explicitly. From our investigations of real-world data, we found that the most commonly used distributions: Normal [37, 2] and Gamma [9, 36] are not a good fit. Meanwhile, approaches using discrete distributions [11, 39] provide an excellent fit, however they do not scale well enough. For other requirements, some methods only support soft time-windows [36, 1], and papers that support time-dependent travel [35, 2, 21, 37] do not cover at least one of our required features: scalability, appropriate service distribution, or hard time-windows. Finally, many solution approaches are evaluated using the Solomon instances [34] or their variations. These instances were not designed for a realistic TRSP: they have a fixed service time (across all technicians and services) and the travel times are Euclidean (symmetric and do not consider traffic). Their difficulty comes from vehicle capacity restrictions, which are absent in our problem.

### 1.2. Contribution and approach

In light of the issues mentioned above, we developed a new approach and a realistic set of instances. One difficulty in modelling realistic service durations comes from their diversity: they range from deterministic to complex multimodal random variables (see Section 3). To address this, we modeled our service durations using phase-type distributions. They generalize a mixture of Erlang distributions and hence can approximate any non-negative distribution to an arbitrary precision [7]. In the context of routing, they were first introduced by [16] and there is a rich and well-developed methodology to fit them to real data [17, 26, 29]. As with most mixture distributions, they become impractical if the representation is too complex [25], however, they are a good and simple fit for distributions in our setting. Furthermore, we can exploit their inherent structure to make our computations more efficient.

In addition to computational improvements, we also developed tools to use phase-type distributions in the presence of hard time-windows. Hard time-windows significantly change the dynamics of the problem [9]; where the main computational cost comes from the conditioning step. For instance, to calculate feasibility, we must iteratively compute the starting time distributions conditioned on hitting all of the previous time-windows. These computations are reasonably straightforward, albeit rather slow, to perform with discrete distributions [10, 39]. However, they can be quite challenging with continuous distributions [9, 19]. One usually has to resort to numerical integration, leading to a significant loss in performance. Our approach avoids numerical integration by leveraging the properties of phase-type distributions to either approximate or avoid computing the conditional distributions. This allows us to accurately calculate the risk and quickly discard infeasible routes.

Time-dependent travel time is another feature that requires careful consideration. Its implementation requires both sophisticated data processing and an efficient computational approach. The methods in the literature either require strict distributional assumptions or do not scale well enough for our purposes [35, 37, 24, 13]. Online mapping services such as Bing and Google Maps provide convenient access to time-dependent traffic data [8], and so we assume it is given as input. Time-dependent travel can also be considered stochastic, however we believe it to be impractical from the data requirements perspective. Specifically, we are not aware of any vendor that provides distribution data for time-dependent travel times. Even if the data was available, its storage and processing needs would render the approach intractable at scale. As a comparison, consider that the stochastic data requirements are much greater than the deterministic case. A 'deterministic' instance with 1000 locations and a 5 day time horizon requires 188 GB of raw time-dependent travel time data, and 440 MB after significant processing and compression. Furthermore, without access to realistic travel time distributions, we are unable to determine the true impact of incorporating stochastic travel. Therefore, we believe that using deterministic time-dependent traffic is

an excellent trade-off between practical methods and high-quality results. Moreover, our approach yields a very accurate approximation with virtually no computational overhead.

A rigorous evaluation of our techniques requires an extensive suite of tests. Rather than trying to extend the instances in [34], we constructed new instances based on real-world data. We randomly selected real geographical locations and obtained the associated time-dependent travel profiles for each pair of locations. Based on our internal data, we generated realistic service time distributions, technician shifts, and request time-windows. These instances were used to evaluate the performance, accuracy and scalability of our techniques. Our experiments in Section 6 show the importance of incorporating risk and time-dependent travel; they highlight the accuracy of our approach; and most importantly, verify that our methodology practically scales to instances of realistic size.

Overall, our contributions can be summarized as follows. We investigated real-world historical data and discovered that service durations are accurately modeled using phase-type distributions. We designed and implemented an efficient 'phase-type based' methodology for evaluating the risk of a technicians route. Our methods capture relevant features of the real world (including time-dependent travel and hard time-windows), and scale well enough to solve instances of practical size. We developed a realistic set of new instances – including service durations, locations, and time-dependent travel profiles, all based on real data. We conducted an extensive computational study, evaluating scalability and the impact of risk and time-dependent travel. We compared our methodology to the natural alternatives that appear in the literature. Finally, we share our tools and instances with the broader research community. These are available at: `https://github.com/microsoft/trsp`.

This paper is organized as follows. The next section formally states our problem and provides a set partitioning formulation. Section 3 discusses the data and how phase-type distributions accurately model service durations. Section 4 presents the probability calculations. Section 5 incorporates time-dependent travel times. We discuss the instances, experiments and computational results in Section 6. Finally, some concluding remarks and future research directions are discussed in Section 7.

## 2. Problem description and model formulation

The objective of TRSP is to find an assignment and ordering of services to technicians that minimizes cost. In our setting, cost is the risk of missing service time-windows or a technician's shift-end. TRSP takes as input the set of technicians $K$, the set of services $V$, and the set of locations $N$. Each technician $k \in K$ has an associated origin $o^k \in N$, destination $d^k \in N$, and shift time-window $[s_k, e_k] \subset \mathbb{R}$. Each service $v \in V$ must start within its time-window $[s_v, e_v] \subset \mathbb{R}$ at the location $o^v \in N$. If a technician arrives at a service before the start of its time-window, they must wait (i.e. hard time-windows)[1]. A service may require specialized skills possessed only by some technicians. Valid service/technician pairs $(v, k) \in V \times K$ have a stochastic service duration $X_{v,k}$ (with known distribution). The time to travel between two locations $n$ and $n'$, departing at time $t$, is given by $\tau_{n,n'}(t)$.

### 2.1. IP Formulation

The following is an extended IP formulation of TRSP. Let $\mathcal{R}_k$ be the set of all feasible routes for the technician $k \in K$, i.e., each route $r \in \mathcal{R}_k$ visits $|r|$ services. Let $c_{kr}$ be the risk associated with route $r \in \mathcal{R}_k$, serviced by technician $k \in K$. We write $v \in r$ to mean that service request $v \in V$ is covered by route $r$. Let $x_{kr}$ be a binary decision variable with value 1 if route $r$ is performed by technician $k$.

[1] Hard time-windows in the stochastic setting might be considered a misnomer. Only the start of the time-window can be enforced, that is, technicians can be late with some probability. Soft time-windows allow a technician to start early with a penalty.

$$\min \sum_{k\in K}\sum_{r\in\mathcal{R}_k} c_{kr}x_{kr}$$

subject to

$$\begin{aligned}
\sum_{k\in K}\sum_{r\in\mathcal{R}_k\,:\,v\in r} x_{kr} &\le 1, && \forall v\in V, && (1)\\
\sum_{r\in\mathcal{R}_k} x_{kr} &\le 1, && \forall k\in K, && (2)\\
\sum_{k\in K}\sum_{r\in\mathcal{R}_k} |r|x_{kr} &= |V|, && && (3)\\
x_{kr} &\in \{0,1\}, && \forall k\in K, r\in\mathcal{R}_k. &&
\end{aligned}$$

Constraint (1) ensures that services are not visited more than once. Constraint (2) makes sure that each technician is assigned to at most one route. Constraints (3) and (1) together ensure that exactly $|V|$ services are completed.

### 2.2. Risk definition

What exactly do we mean by risk? Out of the many useful risk measures, we use two relatively common definitions as seen in the literature [27]. The first is expected tardiness, that is, the sum of expected lateness along the route. The second is the probability of route failure (or probability of infeasibility), i.e., the probability that some time-window is violated. To specify these precisely, we must define them as functions of the service starting-times on a route. Consider the route $r = (o^k, v_1, v_2, \ldots, v_n, d^k)$ serviced by technician $k$. Notice that the services are associated with indices $i \in \{1, 2, \ldots, n\}$, while indices 0 and $n+1$ correspond to the technician's origin and destination depots.

Let $X_i$ be the stochastic service duration at index $i$, i.e., $X_0 = X_{n+1} = 0$, and $X_i = X_{v_i k}$. Let $a_i$ and $b_i$ be the earliest and latest service start-times at index $i$, i.e., $a_0 = a_{n+1} = s_k$, $a_i = s_{v_i}$, $b_0 = b_{n+1} = e_k$, and $b_i = e_{v_i}$ for $i \in \{1, 2, \ldots, n\}$. Then, the service start-time at index $i$ is defined recursively as:

$$S_i = \begin{cases} a_0 & \text{if } i = 0 \\ \max\left(S_{i-1} + X_{i-1} + \tau_{i-1,i}\left(S_{i-1} + X_{i-1}\right),\ a_i\right) & \text{if } i \in \{1, \ldots, n+1\} \end{cases} \tag{4}$$

The technician leaves the depot at time $S_0 := a_0$, i.e., the start time of the first 'service'. They arrive at service $i$ after starting the previous service at $S_{i-1}$, performing the service $X_{i-1}$, and then traveling $\tau_{i-1,i}\left(S_{i-1} + X_{i-1}\right)$. If the technician arrives at $i$ before the earliest allowable start time $a_i$, they must wait and start the service at $a_i$.

Using this notation, our chosen risk measures are $\mathbb{E}\left[\sum_{i=1}^{n+1}\left(S_i - b_i\right)_+\right]$ and $\mathbb{P}\left(\cup_{i=1}^{n+1} S_i > b_i\right)$ respectively. Note that we do not differentiate between a technician being late to a service or working overtime (although it is a straightforward extension).

## 3. Risk evaluation

To evaluate Equation 4, we first need to specify the $X_i$ distributions. A good model must accurately represent real-world service durations, and ideally be computationally easy to work with, i.e., analytical convolutions, translations, and maximums with a constant.

### 3.1. Modeling historical data

In practice, service durations are quite varied and complex. Figure 1 illustrates historical service durations observed by our affiliates. Each histogram corresponds to a single technician performing one type of service request; the data is self-reported via personal electronic devices. Most service durations exhibit patterns seen in histograms (A) and (B). Their distributions are positively

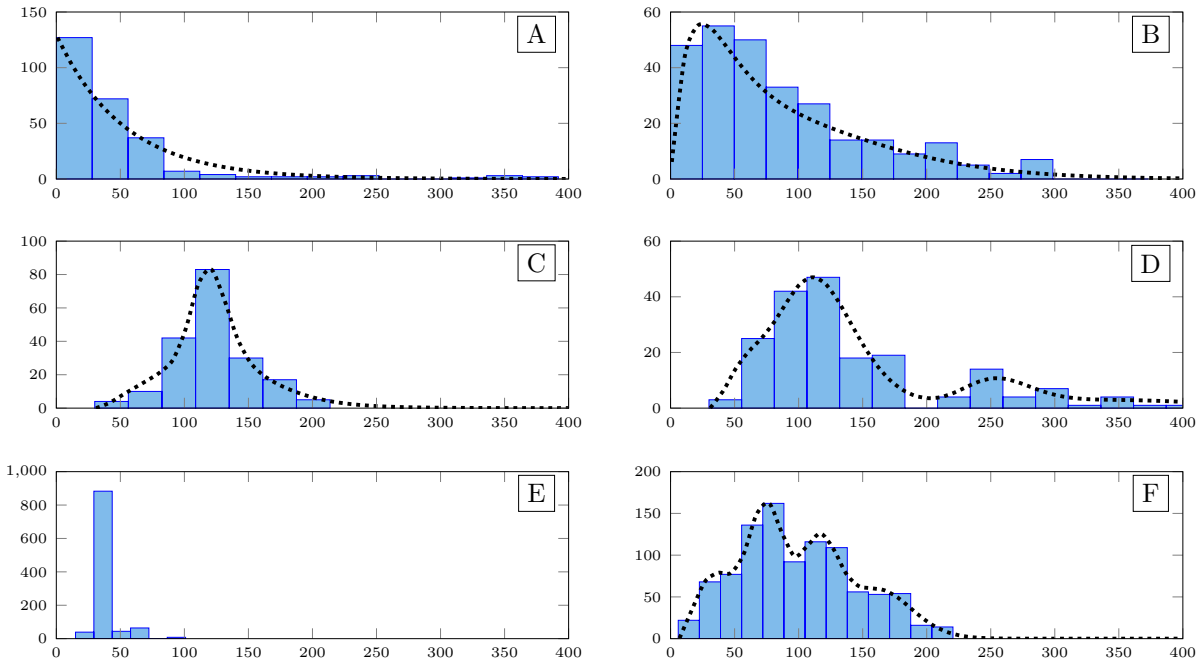



**Figure 1** **Examples of service durations from historical data, and densities of fitted shifted phase-type distributions. The vertical axes correspond to the number of requests, the horizontal axes correspond to service durations in minutes.**

skewed, have relatively significant variance, but the tails are not too heavy. They can be well modelled by Erlang distributions with appropriate parameters. However, many services are much more complicated. For instance, both unimodal (C) and multimodal (D) distributions occur in the data. Multimodal service durations often correspond to problems with multiple possibilities for an underlying cause, which cannot be diagnosed until technicians reach the location. This leads to very complicated distributions for some types of requests. These requests are in contrast to other services, whose duration can be predicted quite reliably. For instance, some service durations are almost deterministic (E), while others have distributions with a very large support (F).

This leaves us with a big problem. Even ignoring time-dependent travel and hard time-windows (i.e., taking a maximum with a constant), we still need to do convolutions and translations. The distributions that behave well under these operations (e.g., Gaussians or Gamma with a fixed scale parameter) do not fit our data. Discrete distributions are a natural and accurate option (if one can avoid numerical issues), however they are typically too slow for a practical solution. Using simulations is another natural approach, although they take far too long to converge to a reasonable level of accuracy. In light of these issues, phase-type distributions [23] are an excellent fit for our requirements.

### 3.2. Phase-type distributions

A phase-type distribution is formally defined as the distribution of the time to absorption in a Continuous-Time Markov Chain of dimension $m+1$, where one state is absorbing and the remaining $m$ states are transient [4]. It is uniquely given by an $m$ dimensional (row) vector $\boldsymbol{\alpha}$ and an $m \times m$ matrix $\mathbf{T}$. The vector $\boldsymbol{\alpha}$ can be interpreted as the initial probability vector among the $m$ transient states, while the the matrix $\mathbf{T}$ is the infinitesimal generator matrix among the transient states. For a given representation $X \sim (\boldsymbol{\alpha}, \mathbf{T})$, the generator matrix for the CTMC is:

$$T = \begin{pmatrix} \mathbf{T} & \mathbf{t} \\ \mathbf{0} & 0 \end{pmatrix}$$

where $\mathbf{t} = -\mathbf{T}\mathbf{1}$. To account for possible non-zero minimum values, we will be working with shifted phase-type distributions, i.e., distributions of the form $Y = y + (\boldsymbol{\alpha}, \mathbf{T})$, where $y$ is a real number and $X \sim (\boldsymbol{\alpha}, \mathbf{T})$ is a phase-type distribution.

One can view phase-type distributions as mixtures of convolutions of exponential distributions. Therefore, they can approximate any distribution with a non-negative support to arbitrary accuracy [7]. There are multiple approaches and tools used to fit phase-type distributions to empirical data. The interested reader can find a good overview in the survey by [26]. For our tests, we used Hyperstar [29], an EM-based graphical utility – some example fits can be seen in Figure 1. As mentioned in [16], shifted phase-type distributions have closed form expressions for convolutions, moments, and probability calculations. The explicit results are covered in their supplemental material. A more in-depth treatment can be found in [4]. For the sake of completeness, we cover them in the Appendix. In addition, we would like to point out a simple property that has not been directly mentioned by other sources: shifted phase-type distributions are closed under maximums with a constant. That is, given $X \sim x + (\boldsymbol{\alpha}, \mathbf{T})$ and $y \in \mathbb{R}$ then

$$Y = \max(X, y) \sim \max(x, y) + (\boldsymbol{\alpha}^\intercal \exp(\mathbf{T}(y - x)_+), \mathbf{T}).$$

**Example** (Expected tardiness)**.** Suppose we would like to compute the expected tardiness of the route $r$. Assume that in a preprocessing step, we fit service durations as shifted phase-type distributions $X_i = x_i + (\boldsymbol{\alpha}_i, \mathbf{T}_i)$. Additionally, let's assume that our travel times are constant, i.e., $t_{i-1} := \tau_{i-1,i}(t)$ for all $t \in \mathbb{R}$. We will relax this assumption and introduce time-dependent travel times in Section 5. Evaluating the first few iterations of Equation 4 yields:

$$\begin{aligned} S_0 &= a_0, \\ S_1 &= \max(a_0 + t_0, a_1), \\ S_2 &= \max(X_1 + S_1 + t_1, a_2). \end{aligned}$$

Notice that these operations are all closed within the shifted phase-type distribution, i.e., $S_0$, $S_1$, and $S_2$ are all shifted phase-type distributions. Continuing in this manner, it can be shown that all service start times, $S_i$, are phase-type distributions. Furthermore, for a given shifted phase-type $S_i \sim s_i + (\boldsymbol{\alpha}_i, \mathbf{T}_i)$, we have:

$$\mathbb{E}\left[(S_i - b_i)_+\right] = \mathbb{E}\left[\max(b_i, S_i)\right] - b_i,$$

where $\max(b_i, S_i)$ is also a shifted phase-type distribution. By linearity of expectation, the expected tardiness of the route is

$$\begin{aligned} \mathbb{E}\left[\sum_{i=1}^{n+1} (S_i - b_i)_+\right] &= \sum_{i=1}^{n+1} \mathbb{E}\left[(S_i - b_i)_+\right] \\ &= \sum_{i=1}^{n+1} \mathbb{E}\left[\max(b_i, S_i)\right] - b_i, \end{aligned}$$

which can be evaluated in closed form.

### 3.3. Efficient computation

In their paper, [16] proposed the use of phase-type distributions to model complex service and travel time distributions (without time-dependent travel or hard time-windows). They evaluated how well this approach compares to normal approximation and a limited number of simulation runs for a variety of proposed service durations, especially heavy tailed ones. Our focus is different. We want to exploit their potential to fit our service durations efficiently and make them usable in large-scale production systems. We would like to make a few qualifying remarks why this is possible. The phase-type distributions' potential for representation comes from the fact that they

are mixture distributions. However, the computational tractability of a mixture distribution relies heavily on the complexity of the representation [25]. If the representation is too complex, the matrix parameter $\mathbf{T}$ become large, dense and ill-conditioned. One of the reasons phase-type distributions work well for our purposes is that they represent most of our data very compactly. As we mentioned before, many service durations we observed can be closely approximated by an Erlang distribution or a mixture of a few Erlang distributions. This leads to a good fit with small bidiagonal generator matrices and sparse initial probability vectors. Furthermore, a lot of our service durations have relatively large coefficient of variation. This leads to small shape parameters in the fitted Erlangs, and makes the dimension of the matrix $\mathbf{T}$ very manageable. Together, these observations guarantee that most of our phase-type distributions $X_i$ are low-dimensional, sparse, and almost diagonal – making them amenable to efficient computations.

Although phase-type distributions have closed form expressions of our calculations, this does not necessarily make them tractable. The tractability comes from carefully exploiting the structure of our specific phase-type distributions to speed up the numerical linear algebra. Most of our operations deal with sparse matrices with a known sparsity structure. To name a few properties, our generator matrices are necessarily upper triangular, the diagonal element dominates off-diagonal elements, and non-zero elements in the top right quadrant of $\mathbf{T}$ can only occur as part of a dense column. Knowing these features allows us to speed up low level operations by exploiting the underlying compressed sparse column (CSC) representation. For instance, we can perform linear solve purely by backward substitution (i.e., without requiring any decompositions); our convolution operations are accelerated significantly by explicitly performing array splicing techniques to combine CSC matrices rather than relying on generic matrix constructors. Furthermore, knowing the dominating terms in the matrix allows us to avoid and correct for a lot of numerical issues.

Exploiting the structure of our matrices is most important for matrix exponentiation. Calculating probabilities, conditional moments (see Section 4), and taking maximums requires the computation of a matrix exponential and vector product of the form $e^{\mathbf{T}x}\mathbf{1}$. In general, evaluating a matrix exponential is a very expensive and numerically unstable operation. A wide variety of methods have been proposed over last few decades [20], none of them are adequate in every situation and most require very careful implementation and tuning. However, in our setting, we only need to perform this operation on very sparse, acyclic CTMC generator matrices. This allows us to adapt a very efficient approach developed by [32] to evaluate the evolution of a general CTMC. It is a Krylov subspace based method, which is both numerically stable and vastly superior to general purpose algorithms. It is fundamental in making our whole approach practical. Our C++ implementation (with C# and Python wrappers) of the shifted phase-type distribution is provided online with our supplementary material.

Finally, we would like to point out the limitations of this approach. If most service distributions are very complex and irregular, one should seek a different methodology. For instance, if the distributions are very complex but tightly bounded, perhaps a robust methodology, e.g., [22], would be more appropriate. Although phase-type or indeed any mixture distributions can fit complex distributions, the computational cost would render such approaches impractical. If, on the other hand, service distributions have very small variances or are tightly concentrated around few support points, then one should utilize discrete distributions. It would be an interesting future research area to examine a situation where service distributions have both large and small (but non-zero) supports. We did not pursue this avenue, as the situation did not arise in our instances.

## 4. Probability of infeasibility

In the previous section, we discussed how to compute expected tardiness. We obtained an exact closed-form expression for this metric by exploiting the properties of phase-type distributions and the linearity of expectation. Evaluating our second metric, the probability of infeasibility, is significantly more challenging. Notice that the random variables for service start-times are not independent, i.e., the delay of one service affects the start-time of future services along the route.

Although we can obtain the exact distribution for each service start-time $S_i$, there is no (simple) expression for the joint distribution $\mathbf{S} = (S_0, S_1, \ldots, S_n, S_{n+1})$. Therefore, the exact closed-form expression of $\mathbb{P}\left(\cup_{i=1}^{n+1} S_i > b_i\right)$ requires more sophisticated machinery.

### 4.1. Infeasibility Tracking

Our approach relies on two key ideas. First, we wish to keep track all possible sources of infeasibility arising from the joint distribution $\mathbf{S}$. Second, we observe that while the joint distribution for $\mathbf{S}$ is difficult to express, the conditional distribution for each service start-time $S_i \,|\, \{S_{i-1} > b_{i-1}, \ldots, S_1 > b_1\}$, with $i \in \{1, \ldots, n+1\}$, is much more straightforward. Specifically, we claim that they can be efficiently represented with a minor extension to the shifted phase-type distribution.

Let $Z_i$ be the event $\{S_i > b_i\}$. By applying the inclusion-exclusion principle, we obtain the following:

$$\mathbb{P}\left(\bigcup_{i=1}^{n+1} Z_i\right) = \sum_{k=1}^{n+1} \left[ (-1)^{k-1} \sum_{\substack{I \subseteq \{1,\ldots,n+1\} \\ |I|=k}} \mathbb{P}\left(\bigcap_{i \in I} Z_i\right) \right] = \sum_{i=1}^{n+1} \left[ \sum_{\mathcal{Z} \in \mathcal{P}(\{Z_1,\ldots,Z_i\})} (-1)^{|\mathcal{Z}|-1} \, \mathbb{P}\left(\bigcap_{Z \in \mathcal{Z}} Z\right) \right],$$

where $\mathcal{P}(\cdot)$ is the powerset operator.

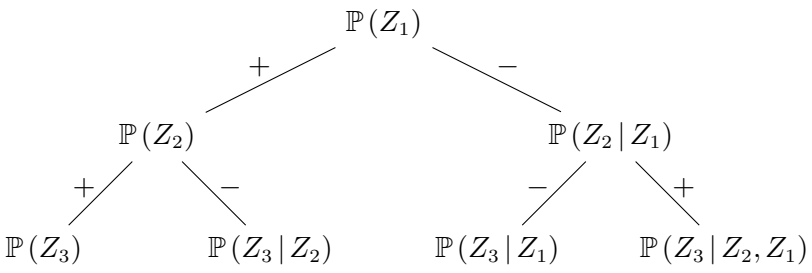
Since each service start-time $S_i$ is dependent on its route history, recursively conditioning on the past simplifies our distributions considerably. We use the notation $\mathcal{Z}_{(j)}$ for the $j$th-smallest item in the set $\mathcal{Z} \subseteq \{Z_1, \ldots, Z_n\}$. For example, with $\mathcal{Z} = \{Z_3, Z_5, Z_6\}$, we have $\mathcal{Z}_{(2)} = Z_5$. Then our metric can be expressed as follows:

$$\begin{aligned}
\mathbb{P}\left(\bigcup_{i=1}^{n+1} Z_i\right) &= \sum_{i=1}^{n+1} \left[ \sum_{\mathcal{Z} \in \mathcal{P}\left(\{Z_1,\ldots,Z_{i-1}\}\right)} (-1)^{|\mathcal{Z}|} \; \mathbb{P}\left(Z_i, \bigcap_{Z \in \mathcal{Z}} Z\right) \right] \\
&= \sum_{i=1}^{n+1} \left[ \sum_{\mathcal{Z} \in \mathcal{P}\left(\{Z_1,\ldots,Z_{i-1}\}\right)} (-1)^{|\mathcal{Z}|} \; \mathbb{P}\left(Z_i \,\middle|\, \bigcap_{Z \in \mathcal{Z}} Z\right) \prod_{j=1}^{|\mathcal{Z}|} \mathbb{P}\left(\mathcal{Z}_{(j)} \,|\, \mathcal{Z}_{(j-1)}, \ldots, \mathcal{Z}_{(1)}\right) \right].
\end{aligned}$$

We illustrate this with a small example ($n = 2$):

$$\begin{aligned}
\mathbb{P}\left(\cup_{i=1}^{3} Z_i\right) = {} & \mathbb{P}(Z_1) \\
& + \mathbb{P}(Z_2) - \mathbb{P}(Z_2 \,|\, Z_1)\,\mathbb{P}(Z_1) \\
& + \mathbb{P}(Z_3) - \mathbb{P}(Z_3 \,|\, Z_2)\,\mathbb{P}(Z_2) - \mathbb{P}(Z_3 \,|\, Z_1)\,\mathbb{P}(Z_1) + \mathbb{P}(Z_3 \,|\, Z_2,\, Z_1)\,\mathbb{P}(Z_2 \,|\, Z_1)\,\mathbb{P}(Z_1)
\end{aligned}$$

The recursion can be visualized by the following binary tree. Branches to the left simply iterate Equation 4, whereas branches to the right also condition on the 'failure' of its parent, and the coefficients are negated.



See Algorithm 1 for the high-level implementation details of this approach. Although the number of terms in our sum grows exponentially with the number of services, a careful implementation can be made relatively efficient for small to moderate $n$.

**Algorithm 1:** Infeasibility tracking

```
def InfTrack(i, [S_1, ..., S_{2^i}], [p_1, ..., p_{2^i}]):
    q ← 0
    for j ← 1 to 2^i do
        Ŝ_j = max(S_j + X_{i-1} + t_{i-1}, a_i)
        Q̂_j = Ŝ_j | Ŝ_j > b_i
        p̂_j = p_j · ℙ(Ŝ_j > b_i)
        q ← q + (-1)^{h(j-1)} p̂_j      // where h(·) is the Hamming weight
    end
    if i < n then
        return q + InfTrack(i + 1, [Ŝ_1, ..., Ŝ_{2^i}, Q̂_1, ..., Q̂_{2^i}], [p_1, ..., p_{2^i}, p̂_1, ..., p̂_{2^i}])
    end
return q
```

It remains to be shown that the shifted phase-type distribution can be extended to efficiently model the above conditional distributions. We introduce the right-conditional shifted phase-type distribution $Y \sim s + (p, \lambda, \boldsymbol{\alpha}, \mathbf{T})$, with the following properties:

**probability**

$$\mathbb{P}(Y \leq y) = \begin{cases} \lambda \left(1 - p - \boldsymbol{\alpha}^\intercal \exp\left(\mathbf{T}(y-s)\right)\mathbf{1}\right) & y > s \\ 0 & \text{o/w} \end{cases}$$

**conditioning**

$$Y \,|\, Y > b \;\sim\; \max\{s, b\} + \left(p + \frac{\mathbb{P}(Y \leq b)}{\lambda},\, \frac{\lambda}{\mathbb{P}(Y > b)},\, \boldsymbol{\alpha}^\intercal \exp\left(\mathbf{T}(b-s)_+\right),\, \mathbf{T}\right)$$

**addition**

$$X_1 + X_2 \;\sim\; (s_1 + s_2) + \left(p_1,\, \lambda_1,\, \left[\boldsymbol{\alpha}_1,\, (1 - p_1 - \boldsymbol{\alpha}_1^\intercal \mathbf{1})\,\boldsymbol{\alpha}_2\right],\, \begin{bmatrix} \mathbf{T}_1 & \mathbf{T}_1 \boldsymbol{\alpha}_2 \\ \mathbf{0} & \mathbf{T}_2 \end{bmatrix}\right),$$

with $X_1 \sim s_1 + (p_1, \lambda_1, \boldsymbol{\alpha}_1, \mathbf{T}_1)$ and $X_2 \sim s_2 + (\boldsymbol{\alpha}_2, \mathbf{T}_2)$.

### 4.2. Approximations

Since our exact approach has an exponential running time in $n$, it is practical to consider fast approximations. One approach would be to try a simple bound. Notice that our service start-times are associated [12], as they are compositions of non-decreasing functions applied to independent random variables. Therefore, the following identity holds:

$$\mathbb{P}\left(\cup_{i=1}^{n+1} S_i > b_i\right) = 1 - \mathbb{P}(S_1 \leq b_1, \ldots, S_{n+1} \leq b_{n+1}) \leq 1 - \prod_{i=1}^{n+1} \mathbb{P}(S_i \leq b_i) \tag{5}$$

In practice, however, this is rather inaccurate. For example, a simple instance with ten Expo(1) services and duration 1 travel times gives the bound $\mathbb{P}\left(\cup_{i=1}^{n+1} S_i > b_i\right) \approx 0.2 \leq 0.35$. Notice that this bound significantly overestimates the probability of infeasibility.

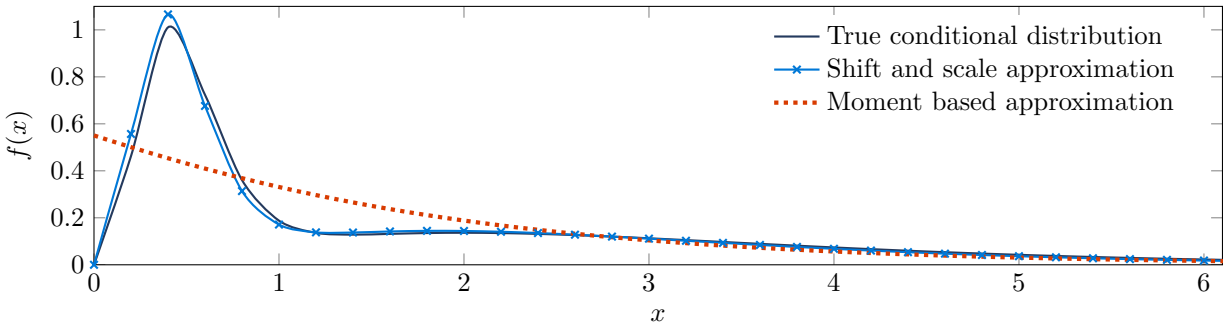



**Figure 2** **Standard moment based approximation of $S_i \,|\, S_i \leq b_i$ [3] vs our scale and shift method**

Clearly we want a better approximation. Specifically, we want an accurate approximation for routes that are likely to succeed and quickly dismiss ones that are not. Equation 5 overestimates the probability due to the lack of conditioning on the past, i.e.,

$$\mathbb{P}(S_i \leq b_i) \;\leq\; \mathbb{P}(S_i \leq b_i \,|\, S_{i-1} \leq b_{i-1}).$$

We correct for this error in the next section, by approximating the conditional distribution.

### 4.3. Shift and Scale

Let $A_i$ be the event $\cap_{j=1}^{i} \{S_j \leq b_j\}$, that is, the event that all services up to and including $i$ started on time. Then, we can write:

$$\mathbb{P}(S_1 \leq b_1, \ldots, S_{n+1} \leq b_{n+1}) = \prod_{i=1}^{n+1} \mathbb{P}(S_i \leq b_i \,|\, A_{i-1}).$$

Using Equation 4, the (time-independent) distribution of $S_i \,|\, A_{i-1}$ is given by

$$S_i \,|\, A_{i-1} = \max(S_{i-1} \,|\, A_{i-1} + X_{i-1} + t_{i-1},\, a_i).$$

If $S_{i-1} \,|\, A_{i-1}$ was a phase-type distribution, the computation would be simple. It is not, but we can approximate it by one. An approach commonly used in the literature is the method of moments [9]. Let $X \sim (\boldsymbol{\alpha}, \mathbf{T})$, then the first two conditional moments are:

$$\mathbb{E}[X \,|\, X \leq x] = \frac{\mathbb{E}[X] - \boldsymbol{\alpha}^\intercal \exp(\mathbf{T}x)\,(x\mathbf{I} - \mathbf{T}^{-1})\,\mathbf{1}}{\mathbb{P}(X \leq x)},$$

$$\mathbb{E}\left[X^2 \,|\, X \leq x\right] = \frac{\mathbb{E}[X^2] - \boldsymbol{\alpha}^\intercal \exp(\mathbf{T}x)\left[(x\mathbf{I} - \mathbf{T}^{-1})^2 + \mathbf{T}^{-2}\right]\mathbf{1}}{\mathbb{P}(X \leq x)}.$$

If we were approximating the distributions with Gaussians, this is all the information we would need. However, with phase-type distributions, there is a considerable amount of freedom in choosing the parameters to match the given moments. The typical approach [3] is not appropriate in our situation, however, our method can derive a good fit (see Figure 2) under certain assumptions. Specifically, we assume that the probability of missing a time-window is low (otherwise the route is likely discarded). Therefore the distribution of $S_i$ is very similar to the distribution of $S_i \,|\, S_i \leq b_i$, and we can exploit its structure. Our approach (see Algorithm 2) matches moments by shifting the distribution and scaling the rate matrix. Suppose $S_i \sim s + (\boldsymbol{\alpha}, \mathbf{T})$, we wish to find $\gamma$ and $\theta$ in $\hat{S}_i \sim s + \gamma + (\boldsymbol{\alpha}, \theta\mathbf{T})$ such that:

$$\mathbb{E}\left[\hat{S}_i\right] = \mathbb{E}[S_i \,|\, S_i \le b_i] =: \hat{m}_1 + s$$
$$\mathbb{E}\left[\hat{S}_i^2\right] = \mathbb{E}[S_i^2 \,|\, S_i \le b_i] =: \hat{m}_2 - \hat{m}_1^2 + (s + \hat{m}_1)^2$$

It is straightforward to show that with $m_1 := \mathbb{E}[S_i - s]$ and $m_2 := \mathbb{E}[(S_i - s)^2]$ we have:

$$\theta = \sqrt{\frac{m_2 - m_1^2}{\hat{m}_2 - \hat{m}_1^2}}, \qquad \gamma = \hat{m}_1 - \frac{m_1}{\theta}.$$

**Algorithm 2:** Shift and Scale

```
def ShiftScale():
    p ← 1
    Ŝ_0 = a_0
    for i ← 1 to n + 1 do
        S_i = max(Ŝ_{i−1} + X_{i−1} + t_{i−1}, a_i) ∼ s_i + (α_i, T_i)
        p ← p · P(S_i ≤ b_i)
        Ŝ_i ∼ s_i + γ + (α_i, θT_i)
    end
return 1 − p
```

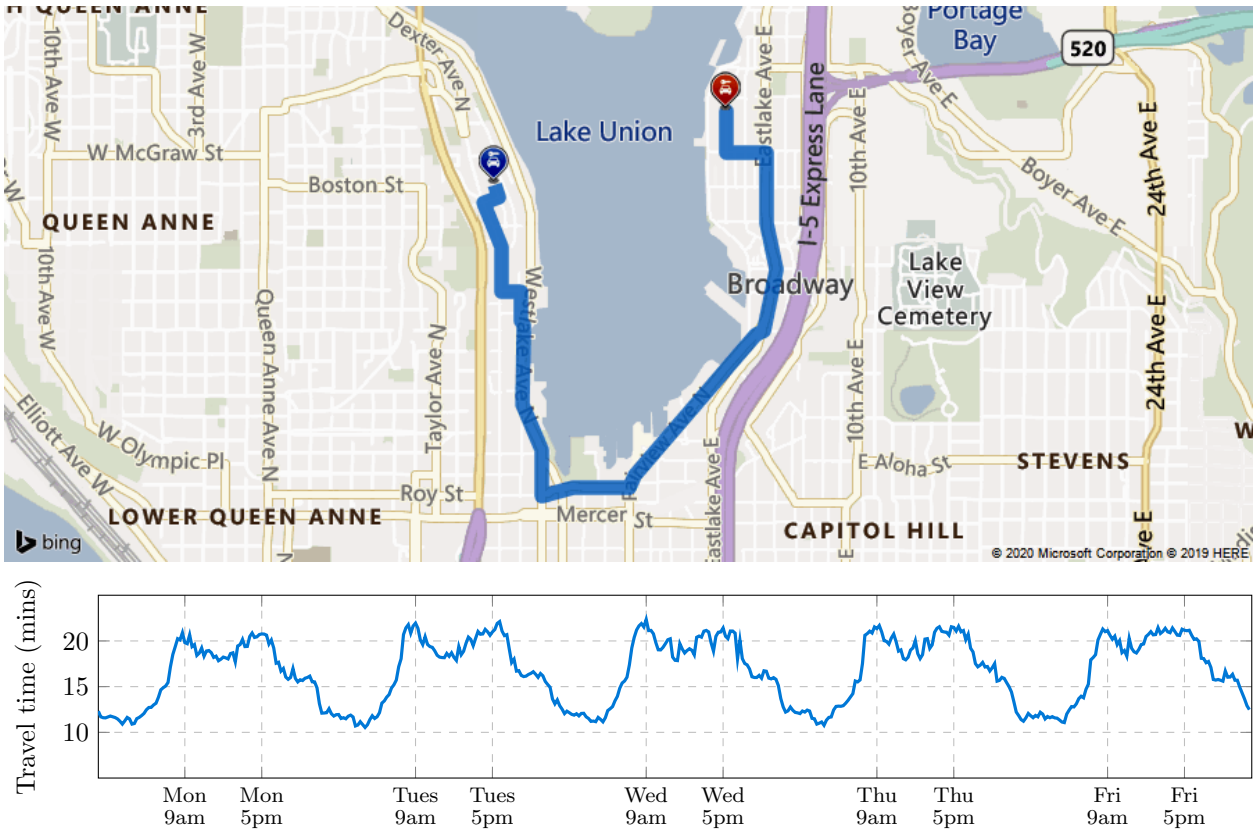



**Figure 3** **The time-dependent traffic profile for an example route.**

## 5. Time-dependent travel

In the real-world, traffic fluctuates throughout the time-of-day and day-of-week. Thankfully, large-scale internet mapping services (like Bing and Google maps) can provide an estimate of time-dependent travel times between any two points [8]. These estimates are based on a variety of sources including GPS traces, historical data, and live user reports. An example route and its associated traffic profile is shown in Figure 3.

Traffic follows a complex periodic pattern, and incorporating this into our service start-time distributions is challenging. Recall from Equation 4 the travel time: $\tau_{i-1,i}(S_{i-1} + X_{i-1})$. By our assumptions on traffic, this is a continuous piecewise-linear function on a phase-type random variable, and returns a random variable with a piecewise phase-type distribution. That is, for $i \geq 1$, Equation 4 can be written as:

$$\hat{S}_i^k = \max\left\{\left(1 + c_i^k\right)\left(S_{i-1} + X_{i-1}\right) + d_i^k,\, a_i\right\},$$

such that

$$S_i = \sum_{k=1}^{N_i} \hat{S}_{i-1}^1 \,|\, C_i^{k-1} \leq \hat{S}_{i-1}^1 < C_i^k, \tag{6}$$

with $N_i$ piecewise segments, and parameters $c_i$, $d_i$, and $C_i$. The FIFO property ensures that every piecewise segment has a slope greater than negative one (i.e., $1 + c_i^k \geq 0$ for all $k \in \{1, \ldots, N_i\}$). Phase-types are closed under the multiplication of a positive constant, thus each segment is a valid phase-type distribution.

The recursive conditional truncation in Equation 6 can be handled exactly by using a similar approach to infeasibility tracking, described in Section 4.1. However, this exact formulation is completely impractical for detailed traffic profiles, since the number of segments is approximately on the order of $2^{n-1}\sum_{i=1}^{n} N_i$. In our computational study, traffic profiles have up to 480 segments over a business week, and so our naïve implementation of this exact method significantly struggled on routes longer than six locations.

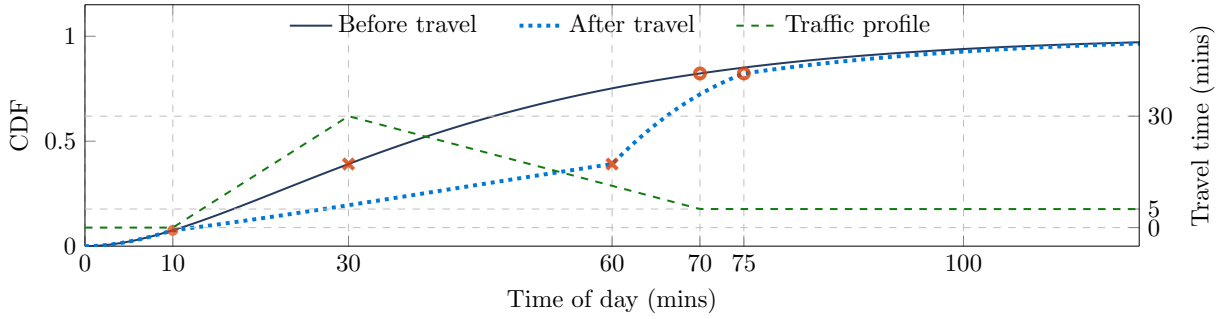



**Figure 4** **The effect of time-dependent travel with a simple piecewise linear function.**

Applying time-dependent travel to a probability distribution can significantly distort its density, as can be seen by Figure 4. By our FIFO assumption, this is a monotonic transformation that expands and contracts piecewise segments. In the figure, see how segment 10 to 30 is stretched from 10 to 60, whereas 30 to 70 is compressed from 60 to 75. Although the transformation can be extreme, in practice it is typically quite reasonable, thus, it is possible to construct an approximation that is both good and efficient, by using linear regression. In doing so, we can reduce our piecewise distribution to a single segment. Let $Y_i = S_i + X_i$, and $Y_i \sim s_i + (\boldsymbol{\alpha}, \mathbf{T})$. We perform linear regression over the traffic interval:

$$\left[s_i + \left(\mathbb{E}\left[Y_i - s_i\right] - 2\sigma_i\right)_+,\ \mathbb{E}\left[Y_i\right] + 2\sigma_i\right],$$

where $\sigma_i$ is the standard deviation of $Y_i$. Figure 5 compares our approximation to the exact distribution with an example. Comprehensive results are shown in the following section.

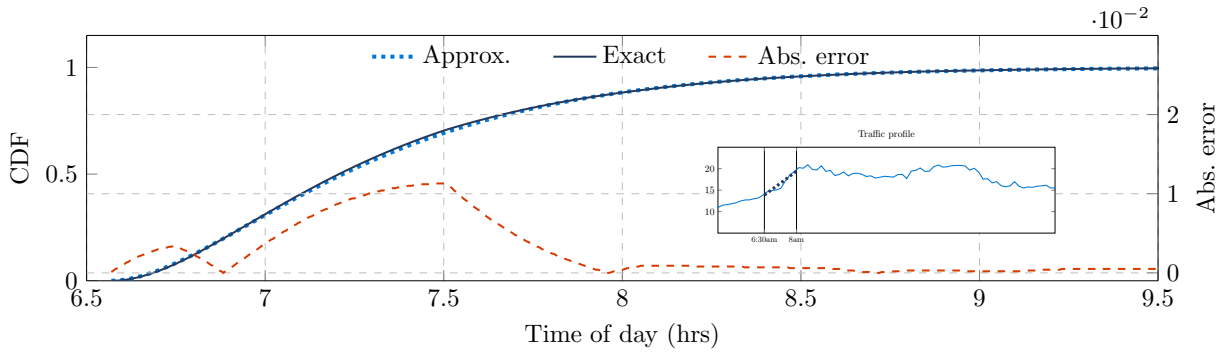



**Figure 5** **Linear regression approximation for time-dependent travel. The absolute error is exaggerated for visualization, and peak hour was chosen for maximum effect.**

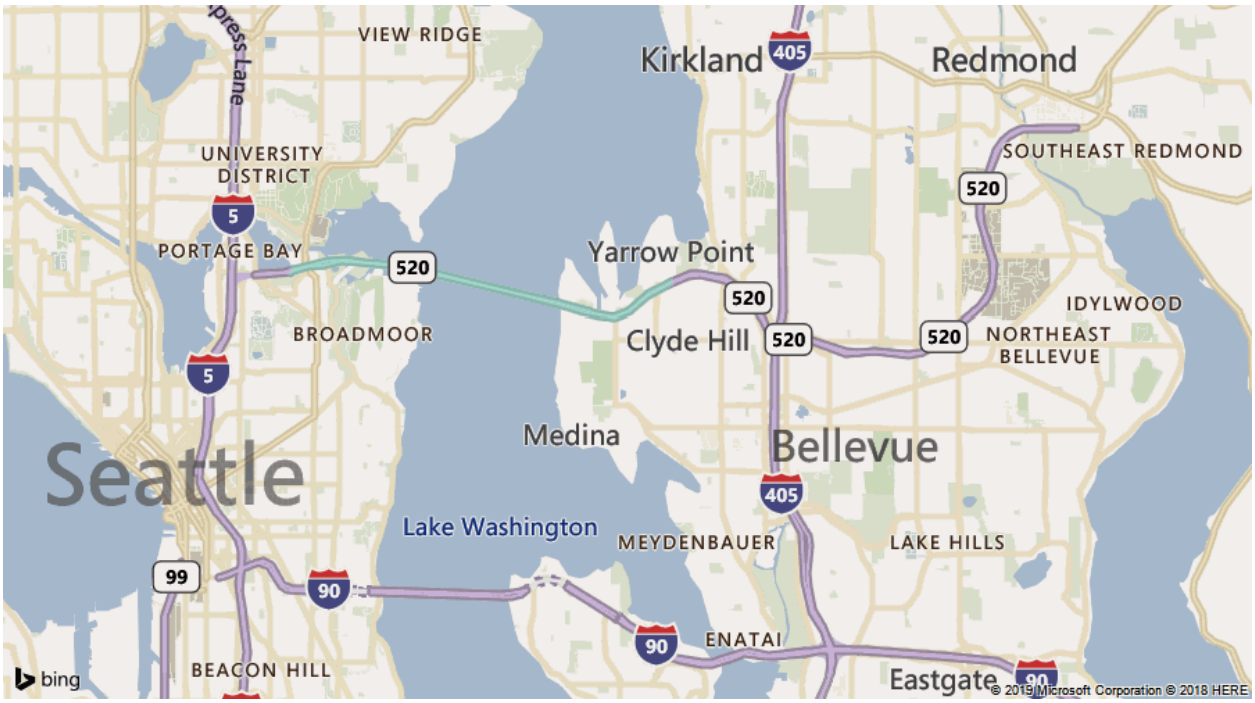



**Figure 6** **Geographic region used for generated instances**

## 6. Computational study

We evaluate our approach against popular alternatives via a large-scale computational study. Our aim is to emulate a realistic environment to verify our effectiveness in practice.

*Instances.* To construct our instances, we sampled 1000 buildings from the Seattle-Bellevue metropolitan area presented in Figure 6. The Bing Maps Distance Matrix API was used to collect traffic profiles between all locations over the first week of July 2019. Technician depot hubs and service requests were randomly assigned to sampled locations. Shift lengths for technicians were fixed at 8 hours, with shift start-time sampled uniformly from 6am to 10am (at 30 minute intervals). Time-windows for services were generated by fixing the time-window length and then sampling the end-time uniformly at random from its earliest completion time to the latest shift end-time (in 15 minute intervals). Instances were split into EASY and HARD categories with probability distributions (0.2, 0.3, 0.5) and (0.5, 0.3, 0.2), respectively, for service time-window sizes of (2, 4, 6) hours.

The distributions for service duration were based from real-world data. Representative examples were extracted and four distinctive shapes were identified: point mass, exponential, erlang, and

multi-modal; chosen with probability distribution (0.01, 0.04, 0.85, 0.1). After choosing a shape, its parameters were randomly adjusted to create new distributions that match a randomly chosen mean and variance. The mean was chosen from a normal distribution such that approximately eight services can be completed per shift. Instances were split into LOW and HIGH variance categories, with the standard deviation uniformly chosen between $[30, 45]$ and $[45, 60]$ minutes respectively. Additionally, instances were also categorized according to the geographical location of the service requests (Rural, Urban, Mixed). Lake Washington was taken to be the boundary between the Rural and Urban categories. For each category, we created 20 instances for each service request sizes: 10, 20, 50, 100, and 200. To test scale we created an additional 20 instances per category with sizes 225, 773, and 999 corresponding to the Rural, Urban, and Mixed locations (with one additional location for technician depot). All instances and parameters are available online.

### 6.1. Probability of Infeasibility

Additional instances were specifically created to compare the methods that calculate the probability of infeasibility. These instances share much of the above construction, but are limited to 200 service requests and 50 technician shifts over a business week. Instances are categorized into LOW and HIGH variance (as defined above), but also *SIMPLE*, *MODERATE*, and *COMPLEX* phase-type shapes with probability distributions (0.7, 0.14, 0.15, 0.01), (0.01, 0.04, 0.85, 0.1), and (0.01, 0.14, 0.15, 0.7) respectively. For each instance we construct a schedule via a simple greedy randomized search heuristic, and replicate this process 30 times for a total of 9000 routes. These routes are used to evaluate our methods.

*Time-independent methods.* We first evaluate methods assuming no traffic. Table 1 shows the 50/90/99 percentiles for both accuracy and computational time. Accuracy (in decimal places) is determined by $-\log_{10}(|p - \mathbb{P}_{\text{InfTrack}}|)$, with $p$ the probability of route failure calculated by a given method. Recall that InfTrack is an exact approach in this setting. The Simulation method is fixed at 2000 iterations; this was chosen to have similar accuracy as ShiftScale for fair comparison. Notice that Discrete [1m] also has similar accuracy, but significantly higher computation time. In particular, notice its *SIMPLE* results: 50% of instances are solved within 0.11ms, however 99% of the instances are solved within 44.02ms – two orders of magnitude slower than ShiftScale. This suggests that there are instances that are incredibly inefficient to discretize. Discrete [10m] has a comparable computation time to ShiftScale, but much worse accuracy. Neither discrete method is numerically stable, i.e., their accuracy drops below 1 decimal place.

Although both InfTrack and ShiftScale methods are comparably fast in our instances, we know that the former has potential to struggle with scale. See Figure 7. The accuracy of ShiftScale decreases with the number of services, however it seems to stabilize around two decimal places. Empirically, the ShiftScale method always over estimates the probability of failure, which is practically desirable in our setting.

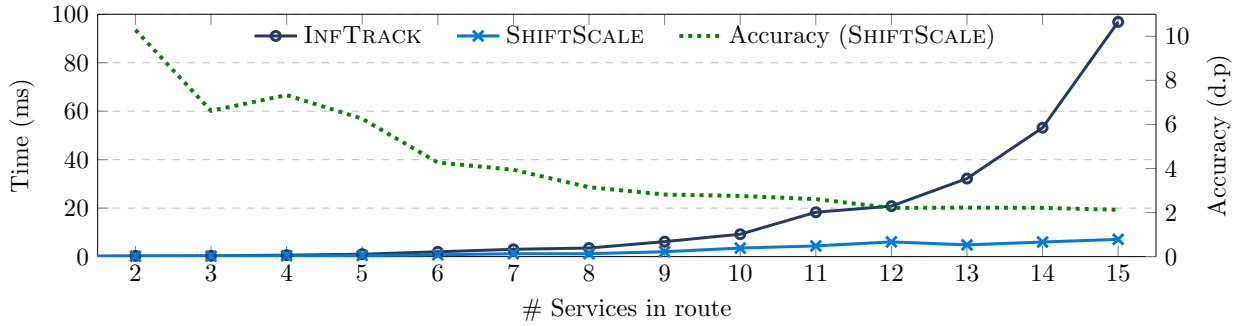



**Figure 7** **The $\mathbf{P_{90}}$ performance of Shift and Scale vs Infeasibility Tracking, as a function of $n$ services along route. Enforces $\mathbb{P}\left(\cup_{i=1}^{n+1} S_i > b_i\right) < 0.05$, to test routes with low probability of failure.**

| Method | Without Traffic | | | | | | Traffic | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Accuracy (d.p.) | | | Time (ms) | | | Accuracy (d.p.) | | | Time (ms) | | |
| | $P_{50}$ | $P_{90}$ | $P_{99}$ | $P_{50}$ | $P_{90}$ | $P_{99}$ | $P_{50}$ | $P_{90}$ | $P_{99}$ | $P_{50}$ | $P_{90}$ | $P_{99}$ |
| *SIMPLE* | | | | | | | | | | | | |
| Discrete [10m] | 2.3 | 0.9 | 0.0 | 0.02 | 0.16 | 0.42 | 2.4 | 0.9 | 0.0 | 0.01 | 0.15 | 0.42 |
| Discrete [1m] | 3.3 | 1.9 | 0.8 | 0.11 | 16.93 | 44.02 | 3.0 | 1.4 | 0.6 | 0.10 | 13.14 | 38.11 |
| Simulation | 3.0 | 2.1 | 1.7 | 1.59 | 2.70 | 4.29 | 1.7 | 0.4 | 0.0 | 1.46 | 2.44 | 5.06 |
| ShiftScale | 4.8 | 2.0 | 1.4 | 0.03 | 0.07 | 0.17 | 3.1 | 1.4 | 0.4 | 0.03 | 0.07 | 0.17 |
| InfTrack | - | - | - | 0.04 | 0.11 | 0.25 | 3.2 | 1.4 | 0.5 | 0.05 | 0.13 | 0.30 |
| *MODERATE* | | | | | | | | | | | | |
| Discrete [10m] | 2.0 | 1.3 | 0.9 | 0.37 | 0.62 | 1.14 | 1.9 | 1.2 | 0.9 | 0.37 | 0.61 | 0.96 |
| Discrete [1m] | 3.0 | 2.2 | 1.5 | 41.57 | 67.41 | 93.18 | 2.3 | 1.4 | 1.1 | 37.47 | 65.38 | 91.65 |
| Simulation | 2.4 | 1.9 | 1.6 | 3.33 | 5.86 | 10.67 | 1.1 | 0.6 | 0.5 | 3.20 | 5.31 | 9.43 |
| ShiftScale | 2.8 | 1.9 | 1.5 | 0.08 | 0.23 | 0.65 | 2.1 | 1.4 | 1.2 | 0.09 | 0.26 | 0.81 |
| InfTrack | - | - | - | 0.12 | 0.38 | 0.94 | 2.1 | 1.4 | 1.1 | 0.14 | 0.42 | 1.00 |
| *COMPLEX* | | | | | | | | | | | | |
| Discrete [10m] | 1.9 | 1.1 | 0.7 | 0.32 | 0.53 | 0.79 | 1.8 | 1.2 | 0.8 | 0.34 | 0.55 | 1.14 |
| Discrete [1m] | 2.8 | 2.0 | 1.0 | 36.65 | 59.64 | 78.88 | 2.2 | 1.4 | 0.9 | 34.06 | 59.83 | 83.37 |
| Simulation | 2.4 | 1.9 | 1.6 | 5.66 | 10.13 | 15.53 | 1.1 | 0.7 | 0.5 | 5.58 | 10.50 | 16.73 |
| ShiftScale | 2.9 | 2.0 | 1.6 | 0.25 | 0.63 | 2.64 | 2.1 | 1.4 | 1.0 | 0.29 | 0.78 | 4.57 |
| InfTrack | - | - | - | 0.32 | 0.90 | 2.30 | 2.1 | 1.4 | 1.0 | 0.38 | 1.09 | 2.60 |

**Table 1 Accuracy vs Speed for the probability calculation methods.**

*Time-dependent methods.* We now incorporate traffic into our methods. InfTrack uses our traffic approximation from Section 5, and thus is not an exact method in this setting. Accuracy is compared against running 10 million simulations per route. Table 1 shows the results. Although they are largely similar to the non-traffic results, we notice that Simulation has significantly less accuracy with traffic – it seems the variability cannot be captured in the limited number of simulation runs. In fact, all methods appear to suffer in accuracy, particularly and surprisingly on the *SIMPLE* instances. However, we suspect that this may be an artifact of our baseline comparison, 10 million simulations might not be sufficient for suitably accurate results. Regardless of these caveats, it is clear that the ShiftScale and InfTrack approaches outperform the other methods in both accuracy and speed.

### 6.2. Expected Tardiness

Many practitioners find expected tardiness to be more interpretable and intuitive than the probability based metrics. In our computational study we evaluate expected tardiness with four types of tests: incorporating risk, comparison to discretization, traffic, and scale. Each test is evaluated within a heuristic optimization framework. Although our methods can be used within an *exact* optimization framework, these typically do not scale to practical sizes.

*Optimization engine.* To perform our evaluation, we use the Adaptive Large Neighbourhood Search (ALNS) metaheuristc introduced by [31], which is popular in many vehicle routing and optimization problems in general. Starting from a feasible state, it iteratively *destroys* and *repairs* the current solution, searching for an improvement. The destroy-repair pair of operators are chosen from a customized set of procedures, and the probability of choosing a pair is based on its success in previous iterations (i.e. the weights are updated adaptively). These iterations are embedded in a simulated annealing framework. Termination is based on a timeout or number of iterations without improvement. Our ALNS implementation has not been heavily optimized. There are many

| Instance | Phase-type | | Deterministic | | | Discrete [10 min] | | | Discrete [1 min] | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | Risk | Iters/s | Risk | Est. | Iters/s | Risk | Est. | Iters/s | Risk | Est. | Iters/s |
| **EASY** | | | | | | | | | | | |
| **LOW** | | | | | | | | | | | |
| Rural | 2.5 | 97.3 | 440.0 | 0.0 | 390.3 | 22.3 | 3.5 | 15.0 | 543.8 | 547.6 | 0.3 |
| Urban | 3.5 | 99.2 | 1191.9 | 0.0 | 425.9 | 54.6 | 18.9 | 15.5 | 2897.5 | 2923.3 | 0.3 |
| Mixed | 3.1 | 88.0 | 1478.8 | 0.0 | 417.9 | 71.7 | 28.0 | 15.1 | 3467.2 | 3501.1 | 0.3 |
| **HIGH** | | | | | | | | | | | |
| Rural | 5.4 | 117.3 | 626.7 | 0.0 | 363.0 | 27.4 | 7.7 | 11.6 | 707.3 | 732.9 | 0.2 |
| Urban | 6.0 | 128.5 | 1564.9 | 0.0 | 344.7 | 187.6 | 174.6 | 11.5 | 2905.0 | 3002.1 | 0.2 |
| Mixed | 5.5 | 109.7 | 1958.3 | 0.0 | 418.2 | 340.9 | 351.6 | 11.6 | 3974.6 | 4128.4 | 0.2 |
| **HARD** | | | | | | | | | | | |
| **LOW** | | | | | | | | | | | |
| Rural | 8.7 | 102.3 | 467.2 | 0.0 | 358.4 | 32.2 | 12.2 | 16.8 | 1125.1 | 1131.6 | 0.3 |
| Urban | 12.4 | 100.6 | 1300.4 | 0.0 | 415.7 | 169.2 | 134.1 | 16.2 | 5675.2 | 5718.7 | 0.3 |
| Mixed | 13.4 | 102.8 | 1643.1 | 0.0 | 397.2 | 537.6 | 511.9 | 16.3 | 7558.3 | 7612.4 | 0.3 |
| **HIGH** | | | | | | | | | | | |
| Rural | 14.1 | 139.0 | 676.8 | 0.2 | 378.2 | 38.2 | 21.6 | 11.5 | 1390.8 | 1427.2 | 0.2 |
| Urban | 16.9 | 135.5 | 1708.6 | 0.2 | 412.5 | 257.4 | 228.8 | 11.5 | 5727.8 | 5862.9 | 0.2 |
| Mixed | 19.6 | 129.1 | 2156.8 | 0.6 | 409.3 | 842.7 | 898.7 | 11.8 | 7800.5 | 7981.2 | 0.2 |

**Table 2 Comparison of results: phase-type vs discretization (without traffic).**

parameters that can be tweaked and more efficient destroy/repair functions that could be used. As the focus of this paper is on risk calculation, we omit further details of the ALNS algorithm.

*Incorporating risk.* We first investigate the value of planning with risk (ignoring traffic). Table 2 shows the results comparing our phase-type approach with a deterministic method. In both approaches expected tardiness is calculated, but the distributions in the deterministic case use a point mass centered at the expected value of service duration. Using average values for service duration sounds reasonable, however the results show that it is a poor proxy. Although it is 2–5 times faster, the associated risk is 50–500 times larger.

*Discretization vs phase-type.* A very common method to calculate risk is by discretization. It is incredibly flexible and simple to understand – however it can have issues with numerical stability and efficiency. In Table 2 we compare our phase-type approach to two different discretization schemes (1 and 10 minute intervals). Note that within our discretization we round up to the nearest interval, as this overestimation greatly helps in optimization. Again we do not consider traffic. From the table, our phase-type approach clearly outperforms (by orders of magnitude) both discrete methods in risk and speed. Although the Discrete [1 min] provides a better estimation of the actual risk (within 2% avg.) than Discrete [10 min] (within 40% avg.), it is, on average, 60 times slower (460 times slower than our phase-type) – for this reason, its use in the optimization procedure yields poor results.

*Incorporating traffic.* We now investigate the effect of traffic when planning under uncertainty. The experimental results are summarized in Table 3. Our time-dependent phase-type approach uses the traffic approximation outlined in Section 5. To calculate the exact risk we use simulation with one million iterations for each route. It is clear from the results that traffic is an important consideration, and our approximation provides an accurate estimate with very little performance impact. Again we see that the discrete approach struggles with both estimation and speed.

*Optimization at scale.* Finally, we evaluate the impact of scale. In particular, we wish to compare the performance of the phase-type, discretization and deterministic methods as instance size grows. The results are in Table 4. As expected, speed seems to be linear in the complexity of the instance. It is quite promising to see very reasonable solutions for our week long schedules with dozens

| Instance | Phase-type | | | TD Phase-type | | | TD Discrete [10 min] | | |
|---|---|---|---|---|---|---|---|---|---|
| | Risk | Est. | Iters/s | Risk | Est. | Iters/s | Risk | Est. | Iters/s |
| *EASY* | | | | | | | | | |
| **LOW** | | | | | | | | | |
| Rural | 403.3 | 2.5 | 97.3 | 5.6 | 5.5 | 78.4 | 29.5 | 8.1 | 14.2 |
| Urban | 506.4 | 3.5 | 99.2 | 7.6 | 7.5 | 76.9 | 56.9 | 18.4 | 14.5 |
| Mixed | 608.7 | 3.1 | 88.0 | 7.4 | 7.4 | 72.1 | 173.4 | 136.8 | 14.4 |
| **HIGH** | | | | | | | | | |
| Rural | 378.5 | 5.4 | 117.3 | 10.1 | 9.9 | 94.0 | 31.3 | 15.0 | 10.7 |
| Urban | 517.7 | 6.0 | 128.5 | 11.8 | 11.6 | 95.5 | 378.8 | 395.2 | 11.1 |
| Mixed | 619.4 | 5.5 | 109.7 | 10.8 | 10.7 | 83.9 | 1565.6 | 1743.4 | 10.8 |
| *HARD* | | | | | | | | | |
| **LOW** | | | | | | | | | |
| Rural | 364.1 | 8.7 | 102.3 | 17.4 | 17.3 | 80.9 | 42.4 | 24.4 | 15.6 |
| Urban | 483.6 | 12.4 | 100.6 | 28.1 | 28.2 | 78.7 | 395.3 | 379.7 | 15.0 |
| Mixed | 551.1 | 13.4 | 102.8 | 30.7 | 31.0 | 80.2 | 750.2 | 754.7 | 15.1 |
| **HIGH** | | | | | | | | | |
| Rural | 372.2 | 14.1 | 139.0 | 24.7 | 24.4 | 107.1 | 48.8 | 34.4 | 10.9 |
| Urban | 463.6 | 16.9 | 135.5 | 32.7 | 32.8 | 100.1 | 470.0 | 467.0 | 11.2 |
| Mixed | 567.0 | 19.6 | 129.1 | 40.0 | 40.0 | 97.7 | 3274.0 | 3555.8 | 11.4 |

**Table 3 Comparison of results: incorporating traffic.**

| #Services | #Techs | #Days | TD Phase Type | | TD Discrete [10 min] | | TD Deterministic | |
|---|---|---|---|---|---|---|---|---|
| | | | Risk | Iters/s | Risk | Iters/s | Risk | Iters/s |
| 10 | 3 | 1 | 11.8 | 354.94 | 13.7 | 26.69 | 62.5 | 949.88 |
| 20 | 7 | 1 | 9.8 | 106.75 | 12.4 | 9.88 | 124.1 | 315.14 |
| 50 | 3 | 5 | 1.7 | 49.85 | 14.6 | 29.49 | 156.0 | 65.64 |
| 100 | 6 | 5 | 2.1 | 15.28 | 32.1 | 9.00 | 327.4 | 19.48 |
| 200 | 12 | 5 | 1.2 | 3.91 | 70.6 | 2.33 | 679.4 | 5.03 |
| 225 | 15 | 5 | 62.4 | 0.90 | 89.2 | 0.12 | 1995.4 | 3.71 |
| 773 | 55 | 5 | 91.0 | 0.11 | 1803.1 | 0.02 | 7265.6 | 0.31 |
| 999 | 70 | 5 | 108.0 | 0.07 | 8502.2 | 0.01 | 9469.8 | 0.19 |

**Table 4 Comparison of results: large-scale optimization (with traffic).**

of technicians and almost 1000 service requests. In comparison, the deterministic proxy is again extremely inaccurate, and the discrete approximation is way too slow and inaccurate to get good solutions.

## 7. Conclusions

This paper proposes efficient methods of evaluating risk for TRSP in a real world setting. Our methods incorporate both hard time-windows and time-dependent travel. To the best of our knowledge, this is the first non-discretization approach that can tackle all of these issues simultaneously. Our framework can be integrated into different optimization engines, both exact and heuristic based. We provide both the low-level code and high-level bindings to make this integration as easy as possible. We evaluated our methods with real-world data, and used this data to build a set of realistic benchmark instances for TRSP, which we hope will be of use to the larger research community.

Our focus for this paper is practicality, with an emphasis on accuracy. We hope to have adequately demonstrated the importance of risk and traffic with our computational results, and we believe

our approach is useful in practice. In regards to future extensions, there are several interesting options to pursue. We believe that further improvements may be possible for the exact approach with traffic. Furthermore, it would be interesting to incorporate our calculations into an exact optimization framework and assess the scalability.

Another extension could be time-dependent service times. Some maintenance requests take more time the longer one waits to tackle them, e.g. pothole repair. If the increase in service times is linear, we can apply our current approach with minimal changes – otherwise, more sophisticated techniques would be required.

Finally, and perhaps the most practical extension, would be to include our risk calculations into a *real* system. At the moment, we focused on a single objective minimizing risk. Real systems are often multi-objective, and can have stochastic requests, online adjustments, and sophisticated recourse policies. In all these directions one would hope to make a reasonable improvement by using an efficient risk evaluation methodology.

## Acknowledgments

We wish to thank Dr. Ishai Menache for our discussions and original motivation in this topic; the internal product partners within Microsoft that provided additional motivation and real-world data; and Prof. Craig Tovey for his valuable feedback on earlier drafts.

### Appendix A: Properties of phase-type distributions

Phase-type distributions have analytic properties that make them quite attractive for our purposes. Let $X \sim x + (\boldsymbol{\alpha}, \mathbf{T})$, be a shifted phase-type distribution. Trivially, $X + s \sim (x + s) + (\boldsymbol{\alpha}, \mathbf{T})$ is also a shifted phase-type distribution. Multiplying by a non negative constant is also allowed: $s \times X \sim sx + (\boldsymbol{\alpha}, \mathbf{T}/s)$. Maximum of $X$ and a constant $s$ is also a shifted phase-type:

$$\max(X, s) \sim \begin{cases} X, \text{ if } s < x, \\ s + \left(\boldsymbol{\alpha}^{\intercal} \exp\left(\mathbf{T}\left(s - x\right)\right), \mathbf{T}\right), \text{ otherwise,} \end{cases}$$

where $\exp(\mathbf{T})$ is a matrix exponential $\exp(\mathbf{A}) = \sum_{n=0}^{\infty} \mathbf{A}^n / n!$.

$$\mathbb{P}\left(X \leq s\right) = \begin{cases} 1 - \boldsymbol{\alpha}^{\intercal} \exp\left(\mathbf{T}\left(s - x\right)\right) \mathbf{1}, \text{ if } s > x, \\ 0, \text{ otherwise} \end{cases}$$

Finally, we have closed form expressions for moments:

$$\mathbb{E}\left[\left(X - x\right)^k\right] = (-1)^k \, k! \boldsymbol{\alpha}^{\intercal} \mathbf{T}^{(-k)} \mathbf{1}.$$

Shifted phase-type distributions are closed under addition. If we have $X_1 \sim x_1 + (\boldsymbol{\alpha}_1, \mathbf{T}_1)$ and $X_2 \sim x_2 + (\boldsymbol{\alpha}_2, \mathbf{T}_2)$, then $Z = X_1 + X_2$ follows a shifted phase-type distribution $Z \sim (x_1 + x_2) + (\boldsymbol{\beta}, \mathbf{U})$, with the initial probability vector $\beta = (\boldsymbol{\alpha}_1, (\boldsymbol{\alpha}_1)_0 \, \boldsymbol{\alpha}_2)$ and the transition rate matrix:

$$U = \begin{pmatrix} \mathbf{T}_1 & \mathbf{T}_1 \boldsymbol{\alpha}_2 \\ \mathbf{0} & \mathbf{T}_2 \end{pmatrix}$$

## References

[1] Adulyasak Y, Jaillet P, 2015 *Models and Algorithms for Stochastic and Robust Vehicle Routing with Deadlines. Transportation Science* 50(2):608–626, URL `http://dx.doi.org/10.1287/trsc.2014.0581`.

[2] Avraham E, Raviv T, 2020 *The data-driven time-dependent traveling salesperson problem. Transportation Research Part B: Methodological* 134:25 – 40, URL `http://dx.doi.org/10.1016/j.trb.2020.01.005`.

[3] Bobbio A, Horváth A, Telek M, 2005 *Matching three moments with minimal acyclic phase type distributions. Stochastic Models* 21(2-3):303–326, URL `http://dx.doi.org/10.1081/STM-200056210`.

[4] Buchholz P, Kriege J, Felko I, 2014 *Phase-type distributions. Input Modeling with Phase-Type Distributions and Markov Models*, 5–28 (Cham: Springer International Publishing), ISBN 978-3-319-06674-5, URL `http://dx.doi.org/10.1007/978-3-319-06674-5_2`.

[5] Castillo-Salazar JA, Landa-Silva D, Qu R, 2016 *Workforce scheduling and routing problems: literature survey and computational study. Annals of Operations Research* 239(1):39–67, URL http://dx.doi.org/10.1007/s10479-014-1687-2.

[6] Chen X, Thomas BW, Hewitt M, 2016 *The Technician Routing Problem with Experience-Based Service Times. Omega* 61:49–61, URL http://dx.doi.org/10.1016/j.omega.2015.07.006.

[7] Cox DR, 1955 *A use of complex probabilities in the theory of stochastic processes. Mathematical Proceedings of the Cambridge Philosophical Society* 51(02):313, URL http://dx.doi.org/10.1017/S0305004100030231.

[8] Cristian A, Marshall L, Negrea M, Stoichescu F, Cao P, Menache I, 2019 *Multi-itinerary optimization as cloud service. Proceedings of the 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, 279–288, SIGSPATIAL '19 (New York, NY, USA: ACM), ISBN 9781450369091, URL http://dx.doi.org/10.1145/3347146.3359375.

[9] Ehmke JF, Campbell AM, Urban TL, 2015 *Ensuring service levels in routing problems with time windows and stochastic travel times. European Journal of Operational Research* 240(2):539–550, URL http://dx.doi.org/10.1016/j.ejor.2014.06.045.

[10] Errico F, Desaulniers G, Gendreau M, Rei W, Rousseau LM, 2016 *A priori optimization with recourse for the vehicle routing problem with hard time windows and stochastic service times. European Journal of Operational Research* 249(1):55–66, URL http://dx.doi.org/10.1016/j.ejor.2015.07.027.

[11] Errico F, Desaulniers G, Gendreau M, Rei W, Rousseau LM, 2016 *The Vehicle Routing Problem with Hard Time Windows and Stochastic Service Times. EURO Journal on Transportation and Logistics* 1–29, URL http://dx.doi.org/10.1007/s13676-016-0101-4.

[12] Esary JD, Proschan F, Walkup DW, 1967 *Association of Random Variables, with Applications. The Annals of Mathematical Statistics* 38(5):1466–1474, URL https://www.jstor.org/stable/2238962.

[13] Gendreau M, Ghiani G, Guerriero E, 2015 *Time-Dependent Routing Problems: A Review. Computers & Operations Research* 64:189–197, URL http://dx.doi.org/10.1016/j.cor.2015.06.001.

[14] Gendreau M, Jabali O, Rei W, 2016 *50th Anniversary Invited Article—Future Research Directions in Stochastic Vehicle Routing. Transportation Science* 50(4):1163–1173, URL http://dx.doi.org/10.1287/trsc.2016.0709.

[15] Guo Z, Wallace SW, Kaut M, 2019 *Vehicle Routing with Space- and Time-Correlated Stochastic Travel Times: Evaluating the Objective Function. INFORMS Journal on Computing* URL http://dx.doi.org/10.1287/ijoc.2019.0906.

[16] Gómez A, Mariño R, Akhavan-Tabatabaei R, Medaglia AL, Mendoza JE, 2016 *On Modeling Stochastic Travel and Service Times in Vehicle Routing. Transportation Science* 50(2):627–641, URL http://dx.doi.org/10.1287/trsc.2015.0601.

[17] Horvath G, Telek M, 2017 *BuTools 2: a Rich Toolbox for Markovian Performance Evaluation. Proceedings of the 10th EAI International Conference on Performance Evaluation Methodologies and Tools* (Taormina, Italy: ACM), ISBN 978-1-63190-141-6, URL http://dx.doi.org/10.4108/eai.25-10-2016.2266400.

[18] Khalfay A, Crispin A, Crockett K, 2017 *A review of technician and task scheduling problems, datasets and solution approaches. 2017 Intelligent Systems Conference (IntelliSys)*, 288–296, URL http://dx.doi.org/10.1109/IntelliSys.2017.8324306.

[19] Miranda DM, Conceição SV, 2016 *The Vehicle Routing Problem with Hard Time Windows and Stochastic Travel and Service Time. Expert Systems with Applications* 64:104–116, URL http://dx.doi.org/10.1016/j.eswa.2016.07.022.

[20] Moler C, Van Loan C, 2003 *Nineteen Dubious Ways to Compute the Exponential of a Matrix, Twenty-Five Years Later. SIAM Review* 45(1):3–49, URL http://dx.doi.org/10.1137/S00361445024180.

[21] Montero A, Méndez-Díaz I, Miranda-Bront JJ, 2017 *An integer programming approach for the time-dependent traveling salesman problem with time windows. Computers & Operations Research* 88:280–289, URL http://dx.doi.org/10.1016/j.cor.2017.06.026.

[22] Munari P, Moreno A, De La Vega J, Alem D, Gondzio J, Morabito R, 2019 *The Robust Vehicle Routing Problem with Time Windows: Compact Formulation and Branch-Price-and-Cut Method. Transportation Science* 53(4):1043–1066, URL http://dx.doi.org/10.1287/trsc.2018.0886.

[23] Neuts MF, 1994 *Matrix-geometric Solutions in Stochastic Models: An Algorithmic Approach*. Algorithmic Approach (Dover Publications), ISBN 9780486683423.

[24] Ng KKH, Lee CKM, Zhang SZ, Wu K, Ho W, 2017 *A multiple colonies artificial bee colony algorithm for a capacitated vehicle routing problem and re-routing strategies under time-dependent traffic congestion. Computers & Industrial Engineering* 109:151–168, URL http://dx.doi.org/10.1016/j.cie.2017.05.004.

[25] Nguyen HD, McLachlan G, 2019 *On approximations via convolution-defined mixture models. Communications in Statistics - Theory and Methods* 48(16):3945–3955, URL http://dx.doi.org/10.1080/03610926.2018.1487069.

[26] Okamura H, Dohi T, 2016 *Fitting Phase-Type Distributions and Markovian Arrival Processes: Algorithms and Tools*, 49–75 (Cham: Springer International Publishing), ISBN 978-3-319-30599-8, URL http://dx.doi.org/10.1007/978-3-319-30599-8_3.

[27] Oyola J, Arntzen H, Woodruff DL, 2018 *The stochastic vehicle routing problem, a literature review. EURO Journal on Transportation and Logistics* 7(3):193–221, URL http://dx.doi.org/10.1007/s13676-016-0100-5.

[28] Pecin D, Contardo C, Desaulniers G, Uchoa E, 2017 *New Enhancements for the Exact Solution of the Vehicle Routing Problem with Time Windows. INFORMS Journal on Computing* 29(3):489–502, URL http://dx.doi.org/10.1287/ijoc.2016.0744.

[29] Reinecke P, Krauß T, Wolter K, 2013 *Phase-type fitting using hyperstar*. Balsamo MS, Knottenbelt WJ, Marin A, eds., *Computer Performance Engineering*, 164–175 (Berlin, Heidelberg: Springer Berlin Heidelberg), ISBN 978-3-642-40725-3, URL http://dx.doi.org/10.1007/978-3-642-40725-3_13.

[30] Ritzinger U, Puchinger J, Hartl RF, 2016 *A survey on dynamic and stochastic vehicle routing problems. International Journal of Production Research* 54(1):215–231, URL http://dx.doi.org/10.1080/00207543.2015.1043403.

[31] Ropke S, Pisinger D, 2006 *An Adaptive Large Neighborhood Search Heuristic for the Pickup and Delivery Problem with Time Windows. Transportation Science* 40(4):455–472, URL http://dx.doi.org/10.1287/trsc.1050.0135.

[32] Sherlock C, 2018 *Simple, fast and accurate evaluation of the action of the exponential of a rate matrix on a probability vector*. URL http://arxiv.org/abs/1809.07110.

[33] Shi Y, Boudouh T, Grunder O, Wang D, 2018 *Modeling and solving simultaneous delivery and pick-up problem with stochastic travel and service times in home health care. Expert Systems with Applications* 102:218–233, URL http://dx.doi.org/10.1016/j.eswa.2018.02.025.

[34] Solomon MM, 1987 *Algorithms for the Vehicle Routing and Scheduling Problems with Time Window Constraints. Operations Research* 35(2):254–265, URL http://dx.doi.org/10.1287/opre.35.2.254.

[35] Taş D, Dellaert N, van Woensel T, de Kok T, 2014 *The Time-Dependent Vehicle Routing Problem with Soft Time Windows and Stochastic Travel Times. Transportation Research Part C: Emerging Technologies* 48:66–83, URL http://dx.doi.org/10.1016/j.trc.2014.08.007.

[36] 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. European Journal of Operational Research* 236(3):789–799, URL http://dx.doi.org/10.1016/j.ejor.2013.05.024.

[37] Verbeeck C, Vansteenwegen P, Aghezzaf EH, 2016 *Solving the stochastic time-dependent orienteering problem with time windows. European Journal of Operational Research* 255(3):699–718, URL http://dx.doi.org/10.1016/j.ejor.2016.05.031.

[38] Zamorano E, Stolletz R, 2017 *Branch-and-price approaches for the Multiperiod Technician Routing and Scheduling Problem. European Journal of Operational Research* 257(1):55–68, URL http://dx.doi.org/10.1016/j.ejor.2016.06.058.

[39] Zhang J, Lam WHK, Chen BY, 2013 *A Stochastic Vehicle Routing Problem with Travel Time Uncertainty: Trade-Off Between Cost and Customer Service. Networks and Spatial Economics* 13(4):471–496, URL http://dx.doi.org/10.1007/s11067-013-9190-x.