Enough Waiting for the Cable Guy -
Estimating Arrival Times for Service Vehicle Routing

Marlin W. Ulmer        Barrett W. Thomas

February 26, 2017

Abstract

Service providers dispatch vehicles to provide technical services or deliver goods. In many cases, customers must be present when the provider arrives, and customers are given a time window in which the service or delivery will take place. These time windows must often be given at the time of the request. However, at the time of the request, not all requests that will be serviced are known, and thus there is uncertainty in the vehicle’s arrival time. In this paper, we present a method that anticipates future requests and their impact on arrival times in the determination of suitable time windows. The proposed approach provides a state-dependent estimate, relying on a state-space aggregation to account for the nearly infinite state space. We propose an offline method that draws on simulation to learn the values associated with aggregated states. We compare the proposed approach with myopic and conventional planning approaches. Results show that our approach significantly increases the service level relative to the benchmarks. The computational experiments also show that, to achieve the highest level of customer satisfaction, the time windows should be application specific and should depend on the time at which the customer requests service.

Keywords: service routing, stochastic customer requests, time-window assignment

1 Introduction

Everyone hates waiting for the cable guy. In fact, people dislike “waiting for the cable guy” so much that it has spawned its own meme with over two million Google search results. More generally,
customers dislike the wide time windows (TWs), the earliest and latest times at which a service will begin, that service providers and home-attended delivery companies provide customers. While customers are willing to tolerate some amount of wait (Zeithaml et al. 1993), these wide TWs are frustrating for customers who often must take at least a part of a day off work to stay at home and wait for the technician (Ragsdale 2012). For example, one of the co-authors was recently told that the technician flipping the switch for internet service would arrive between 8am and 2pm. While these wide TWs offer home service companies better scheduling and routing flexibility, one study estimated that the economic loss resulting from people waiting for service amounted to $38 billion in 2011 in the United States alone (Ellis 2011). Further, even with wide TWs, companies’ can fail to provide service when they said they would, damaging customer satisfaction and potentially repeat business. In one recent survey, 67% of respondents said that they would not again do business with a company whose technician was even an hour late (Guinn 2016). In some cases, a missed appointment may also lead to penalties for the service provider (Karabasz and Ludowig 2016).

In an effort to alleviate customer frustration, companies have begun to roll out real-time technician tracking applications (Nickelsburg 2016, Zara 2015). Yet, while these new technologies may mean that customers are present when a delivery or technician arrives, the better solution is to combine this real-time tracking with narrower TWs and to make sure that technicians arrive within the given TWs. However, simply providing a more narrow time window is challenging. In addition to the uncertainty in travel and service times, TWs must often be communicated at the time that the customer makes the service request, before all of the requests that will be served on that day are known. The integration of new customers into the planned tours shifts the arrival times of already assigned customers. Hence, dispatchers need to anticipate future requests in the determination of suitable arrival times.

In this paper, we seek to improve the customer experience by improving the width of the TWs that are offered without sacrificing reliability. The key to the problem is to estimate a state-dependent arrival time of a technician at the time of the service request when we do not know all of the requests that the technician will be asked to serve on that day. Suitable TWs can then be proposed based on the arrival time estimate. We call the problem the time window assignment problem with dynamic requests (TWAP). Over the course of the day, customers request service for some future day. We call this the capture phase. The execution of the route takes place during a future execution phase.
For the purposes of this study, we assume that the day and the driver to which the request is allocated are determined exogenously by a given dynamic policy. This policy most notably makes routing and assignment decisions for newly arriving requests. With the day and driver chosen by this policy, the decision maker must communicate to each requesting customer a time window in which the technician will arrive to begin the service. We assume that the requests as well as the travel and service times are stochastic. To determine this time window, we first seek a median arrival time around which we center the time window. To this end, we solve an optimization problem whose objective is to minimize the expected absolute difference between the actual arrival time and the communicated arrival time.

The optimal solution for the TWAP minimizing the expected absolute deviation is to determine the state-dependent median arrival time for each customer. We demonstrate that estimates of the median are unstable and instead focus on estimating mean arrival times. To estimate the expected arrival time of a technician at a customer, we make use of techniques familiar in approximate dynamic programming. First, we use the concept of a state to describe the status of the system at the time of each request. In this way, we can make each arrival time estimate state-dependent. Because of the large number of potential states, we aggregate the state space based on important temporal parameters. This aggregation is motivated by the success of temporally-based aggregation schemes in the dynamic routing literature. With the aggregated state space, we then execute a series of offline simulations that allow us to learn the expected arrival times for a state. To achieve a balance between simulation runs and solution quality, we rely on a dynamic aggregation of the state space. By performing the computation offline, the proposed method is particularly amenable to the need to communicate TWs in real time.

This paper makes several important contributions to the literature. First, we introduce a new model and method valid for a number of applications related to completion time estimation in both dynamic routing and scheduling. Second, we introduce a temporal-aggregation scheme for the state space. This aggregation allows us to develop high-quality state-dependent estimates of the arrival times for a given request and thus provide high quality state-dependent TWs. Notably, our aggregation scheme introduces new measures that allow us to implicitly capture information about customer locations relative to the depot and the other customers in the tour. These measures may also be useful in aggregation used for approximate dynamic programming for dynamic vehicle
routing. Given the challenges associated with estimating the median of the arrival-time distributions
in the case of a large state space, we instead estimate mean arrival times and show that our estimates
of the mean lead to superior TWs. We present an offline approach for developing these estimates,
giving us the ability to communicate TWs in real time. We believe that this is the first offline method
that returns time-dependent arrival time estimates for dynamic vehicle routing problems. In an
extensive computational study based on customer locations in the area around Iowa City, Iowa,
USA, we demonstrate that the proposed method significantly outperforms the benchmarks. Further,
we make two important observations for companies seeking to maximize customer satisfaction by
balancing the width of the time window and the ability to provide on-time service:

1. Applications with shorter expected service times can offer narrower state-dependent time
   windows while still providing on-time service.

2. State-dependent time windows are least accurate for customers who request service early in
   the capture phase. As a result, a company might achieve greatest customer satisfaction using
   a scheme that provides wider time windows to early requesting customers and narrower time
   windows for those who request later in the horizon.

The paper is organized as follows. In §2, we review related literature, focusing on arrival time
estimation in routing and scheduling. In §3, we describe the underlying routing problem, formally
define and model the TWAP. In §4, we present the proposed solution approach and describe two
benchmarks adapted from the literature to which we compare. In §5, we present test instances
derived from the geography of Iowa City, Iowa. We compare the proposed approach and benchmarks
in an extensive computational study in §6. This paper concludes with a summary and an outlook on
future work in §7.

2 Literature

In the following, we review related literature. We first present work in vehicle routing literature
in which dispatchers are required to evaluate and/or communicate a time window or an expected
arrival time. We then present related work from the field of scheduling. We note that the TWAP
itself is not a dynamic routing problem, but it originates from a dynamic routing problem introduced
in Madsen et al. (1996). For a general overview on dynamic vehicle routing problems, the interested reader is referred to Ritzinger et al. (2015).

We classify the existing work in Table 1. The table first characterizes whether or not all the customers are known at the start of the problem. In the problem discussed in this paper, the customers are not all known at the start of the problem. Rather, they are revealed over time. Particularly important in the case of unknown customers is that the time window must be communicated in real time, at the time of the request.

The table then distinguishes between each paper’s model and solution approach. We categorize the models with regard to whether or not the model requires the ordering of customers or tasks and by the source of uncertainty in the model. By ordering, we refer to whether or not new customers or tasks can be routed or scheduled for processing before an existing customer. When customers are unknown, the inclusion of ordering means that new customers can shift the arrival or completion time. We classify the solution approaches with respect to their anticipation. By anticipation, we refer to whether or not the solution approach accounts for uncertainty in future events by incorporating stochastic information. Finally, because in the case of unknown customers we must provide real time feedback to waiting customers, we identify when the main amount of calculation is conducted offline. Offline approaches, such as is used in this paper, perform computations before the need for execution, storing the results either in a lookup table or via some functional form. In contrast, online approaches react to requests and the resulting system state. Online approaches are often challenged to provide responses to customers fast enough to be suitable in many applications or sacrifice solution quality for computation time (Ritzinger et al. 2015).

The literature on arrival time estimation or TW-assignment in context of stochastic dynamic vehicle routing is limited. The most closely related work is that of Madsen et al. (1996). Madsen et al. (1996) consider a problem similar to the one in this paper in which TW-assignments must be given to customers at the time of the request and before all requests are known. Madsen et al. (1996) assumes uniformly distributed customers and presents a method that updates tours with respect to dummy customers located throughout the service area. For tours of up to five customers and TWs of two hours, the presented method is capable of providing customers with TWs that are reliably met. In contrast to this work, Madsen et al. (1996) does not explicitly anticipate future requests and their impact on the arrival times at customers. Further, the linear approximation presented
Table 1: Literature Classification

<table>
<thead>
<tr>
<th>Literature</th>
<th>Problem Model</th>
<th>Solution Approach</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Ordering</td>
<td>Stochasticity</td>
</tr>
<tr>
<td>Agatz et al. (2011)</td>
<td>✓</td>
<td>-</td>
</tr>
<tr>
<td>Jabali et al. (2015)</td>
<td>✓</td>
<td>Travel Times</td>
</tr>
<tr>
<td>Spliet and Gabor (2015)</td>
<td>✓</td>
<td>Demands</td>
</tr>
<tr>
<td>Spliet and Desaulniers (2015)</td>
<td>✓</td>
<td>Demands</td>
</tr>
<tr>
<td>Zhang et al. (2015)</td>
<td>✓</td>
<td>Travel Times</td>
</tr>
<tr>
<td>Homen-de-Mello et al. (1999)</td>
<td></td>
<td>Processing Times</td>
</tr>
<tr>
<td>Hopp and Roof Sturgis (2000)</td>
<td></td>
<td>Workload</td>
</tr>
<tr>
<td>Elmaghraby (2001)</td>
<td>✓</td>
<td>Processing Times</td>
</tr>
<tr>
<td>Bendavid and Golany (2011)</td>
<td>✓</td>
<td>Processing Times</td>
</tr>
<tr>
<td>Ioannou and Dimitriou (2012)</td>
<td></td>
<td>Workload</td>
</tr>
<tr>
<td>Zhao et al. (2014)</td>
<td></td>
<td>Processing Times</td>
</tr>
<tr>
<td>Atan et al. (2016)</td>
<td></td>
<td>Leadtimes</td>
</tr>
<tr>
<td>Yang et al. (2014)</td>
<td>✓</td>
<td>Requests</td>
</tr>
<tr>
<td>Campbell and Savelsbergh (2005)</td>
<td>✓</td>
<td>Requests</td>
</tr>
<tr>
<td>Ehmke and Campbell (2014)</td>
<td>✓</td>
<td>Requests, Travel Times</td>
</tr>
<tr>
<td>Hodgson et al. (1997)</td>
<td>✓</td>
<td>Requests, Processing Times</td>
</tr>
<tr>
<td>Madsen et al. (1996)</td>
<td>✓</td>
<td>Requests</td>
</tr>
<tr>
<td>TWAP and ATW</td>
<td>✓</td>
<td>Requests, Travel and Service Times</td>
</tr>
</tbody>
</table>

as a benchmark in this work mimics the approach presented in Madsen et al. (1996). Our results demonstrate that the approach proposed in this work significantly improves upon an approach such as is presented in Madsen et al. (1996).

In contrast to this work and Madsen et al. (1996), most of the existing work on TW-assignment focuses on TW-assignments when travel times or demand at known customers are uncertain or explores TW-pricing for problems with stochastic requests. The problem presented in Spliet and Gabor (2015) and Spliet and Desaulniers (2015) considers stochastic demands and contains two stages. In the first stage, TWs are communicated before the customer demand is revealed. In the second stage, the demand is realized, and the dispatcher determines a set of routes that serve the customers in their TWs. Spliet and Gabor (2015) and Spliet and Desaulniers (2015) sample scenarios and approximately solve the deterministic mixed-integer program (MIP) via branch-and-price and column generation. Relatedly, using expected demand by zip code and before observing realizations of the requests in a zip code, Agatz et al. (2011) seeks to determine which time slots to offer in each zip code in a service area.

For a known set of customers and stochastic travel times, Jabali et al. (2015) seek to find routes and according TWs. To account for the uncertainty in travel times, they integrate “time buffers” into their solutions. Zhang et al. (2015) address a problem similar to Jabali et al. (2015) for an inventory
routing problem in maritime shipping. Again, the set of customers is known at the beginning of the horizon and disruptions lead to stochasticity in travel times. Similar to Spliet and Gabor (2015) and Spliet and Desaulniers (2015), Zhang et al. (2015) use sample travel time realizations and solve the resulting MIP. Again, time buffers are utilized to avoid deviations.

A similar problem also arises in make-to-order production scheduling. In many cases, companies must supply customers with estimates of when requested work will be completed. These estimates are known as flowtime or due date estimates and can be associated to arrival times in routing. Like the work in this paper, the work in production scheduling is seeking to provide customers with an estimate in real time. In contrast to the work in this paper, the production scheduling literature generally assumes that requests are processed in first-in-first-out order. Thus, the challenges in estimation are due to random processing times and coordination. In our work, the technician’s route evolves over the day and a request’s position in the route is not known until all requests for a day have been routed. Examples of work in due date estimates can be found in Ioannou and Dimitriou (2012) and Hopp and Roof Sturgis (2000). Zhao et al. (2014) study an analogous problem of estimating customer wait time guarantees in a queue.

An analogous problem to the TWAP can be found in production scheduling in the context of estimating the best time to release material into production so as to meet customer-imposed due dates. Hodgson et al. (1997) considers the possibility that new requests will be placed ahead of existing requests. However, unlike in this paper, Hodgson et al. (1997) does not need to communicate the release time to the customer, and thus there is no value in anticipating future customer requests. Requests can simply be rearranged in response to requests.

In contrast to this work, the remainder of the work in material release timing assumes that jobs are served in the order that they are received and thus no consideration to ordering is given. Among these remaining papers, Homen-de-Mello et al. (1999) use an offline simulation approach, like we do in this paper, to determine release times. Other examples include Elmaghraby (2001), Stanfield et al. (2006), Atan et al. (2016). Bendavid and Golany (2011) discusses a similar problem in project scheduling in which estimates of ready times are needed so that subcontractors can be scheduled. While these papers do assume random processing times, they assume that all the jobs are known at the beginning of the horizon.

As an alternative to TW estimation, planning for efficient operations in home delivery, most
commonly grocery delivery, often requires knowing whether or not a customer’s delivery TW can be efficiently and feasibly served given the fleet and other requests, both current and future. Thus, like this work, the work studying attended home delivery deals with dynamic and stochastic customer requests. In contrast to this work, however, most of the work on attended home delivery does not anticipate future requests, but uses online reoptimization of delivery routes on a rolling horizon.

The work in attended home delivery most closely related to our work is Campbell and Savelsbergh (2005). In Campbell and Savelsbergh (2005), customers from a known set of customers subsequently request service. Each potential customer has a known request probability and TW-preferences. When making a request, customers can choose from a set of time slots offered by the service provider. The objective is to maximize the number of accepted customers. Similar to Campbell and Savelsbergh (2005), Yang et al. (2014) seek to estimate the cost of a customer’s given TW-selection and offer incentives to direct the customers to TWs that maximize the firm’s profit. The problems proposed by Campbell and Savelsbergh (2005) and Yang et al. (2014) are related to that in this paper because, in both cases, they must determine whether or not a customer’s requested delivery time will be profitable. Doing so requires estimating how such a customer’s TW-selection will affect the ability to serve current and future, yet unrealized, requests. In both Campbell and Savelsbergh (2005) and Yang et al. (2014), samples of the future are used to approximate the impact of a given request and TW-combination on future profits. Campbell and Savelsbergh (2005) and Yang et al. (2014) use an online approach and a small set of samples, one and 10, respectively, to evaluate the profit of a given request. In contrast, in this paper, we use an offline approach that learns from thousands of samples in generating state-dependent estimates. Ehmke and Campbell (2014) extend Campbell and Savelsbergh (2005) to consider stochastic and time-dependent travel times. Ehmke and Campbell (2014) anticipates with respect to travel times but not with respect to new requests. As we show, the impact of new arrivals is more significant for the TWAP than the impact of stochastic travel and service times.
3 The Time Window Assignment Problem with Dynamic Requests

In this section, we model the TWAP. The TWAP is connected to an underlying dynamic acceptance and routing problem, the dynamic vehicle routing problem with stochastic requests (VRPSR). For the purposes of the TWAP, we assume that this underlying VRPSR is solved exogenously to the TWAP and that the TWAP uses the input of the solution of the VRPSR. In our computational study, we assume this routing and acceptance policy for the VRPSR is a combination of non-selective acceptances and insertion routing, but the policy can be chosen arbitrarily. In the following, we first briefly describe the underlying VRPSR. For a detailed Markov decision process-model of the VRPSR, the interested reader is referred to §A.2 in the Appendix.

3.1 VRPSR

In the VRPSR, customers request service during a period of time known as the capture phase. This phase has a duration of $t_{c_{\text{max}}}$. The service of the requests occurs during an execution phase of duration $t_{s_{\text{max}}}$. We assume that customers are distributed across a known region $A$ and that each requesting customer $i$ requires a random service time $\zeta_i$. Travel between any two customers $i$ and $j$ is random and denoted $d_{ij}$. Upon making a request, some exogenous assignment and routing policy is used to assign the request to a technician. This policy might reflect that technicians serve given delivery zones, such as those described in Ouyang (2007), or might reflect a match between the technician’s skillset and the skills required to complete the request such as occurs in Firat and Hurkens (2012). The tour is built iteratively as the customers request service. The acceptance and routing decision of the VRPSR can be chosen freely as long as the policy rejects a request if the increase in expected tour duration due to the addition of the customer would violate $t_{s_{\text{max}}}$. To accommodate realizations of random travel and service times, in the execution of the VRPSR-tour, we do allow for the maximum tour duration constraint to be violated during the execution of the vehicle tour.
3.2 TWAP

A decision point for the TWAP occurs when a request is added to the existing VRPSR-tour. This interaction between the TWAP and VRPSR is illustrated in Figure 1. The grey in the figure represents operations related to the VRPSR. The blue represents operations of the TWAP. The VRPSR problem feeds information to the TWAP. Both the VRPSR and, in the case of an accepted request, the TWAP return information back to the customers.

We let $n_k$ represent that $k^{th}$ request and $t_k$ the time of the request. The set of all accepted requests including the $k^{th}$ request is $\mathcal{N}_k$. The existing tour including the $k^{th}$ request is given by the vector $\theta_k$. The first and last entries of $\theta_k$ are the depot and each other entry a customer request with $\theta^i_k$ being the customer in the $i^{th}$ position on the tour. The distance between any two customers $\theta^i_k$ and $\theta^j_k$ is given by $d(\theta^i_k, \theta^j_k)$. We then define the decision state as $S_k = (t_k, \mathcal{N}_k, \theta_k)$. Given that we assume continuous time, this resulting state space is infinite.

For an example of the system state, see Figure 2. The square represents the depot $D$, the gray circles the previously assigned customers $\mathcal{N}_k \setminus \{n_k\}$, and the light circle a newly assigned request $n_k$. The arrows indicate the current tour $\theta_k$. In the example, $n_k$ is the sixth assigned customer, i.e., $k = 6$. The current sequence is $\theta_k = (D, n_5, n_2, n_3, n_1, n_k, n_4, D)$.

Requests that are added to the tour must then be immediately given an estimated arrival time $X$ or the according time window $[X - \nu, X + \nu]$ of size $\nu$. This TW is then communicated simultaneously with the VRPSR-decision as shown in Figure 1. The objective of this problem is
to minimize the difference from the estimated arrival time, which is the center point of the TW, and the actual arrival time of the technician to request \( n_k \) given that not all customers who will be served by the tour are known at the time that we must communicate the TW.

Given a state \( S_k \), the dispatcher determines an estimated arrival time \( X(S_k) \) for the new request \( n_k \). While the evolution of the underlying routing problem will affect the arrival time at the customers, the selection of an arrival time affects neither the evolution of the underlying routing problem nor the selected arrival times for previous and subsequent customers. As a result, we can treat each selection individually. Because the actual arrival time \( A(S_k) \) is a random variable dependent of state \( S_k \), the objective for each request is to minimize the expected absolute difference for each state \( S_k \) given as:

\[
\arg \min_{X(S_k) \in \mathcal{X}(S_k)} \mathbb{E} \left( |X(S_k) - A(S_k)| \right). \tag{1}
\]

The optimal solution of Equation (1) is the median \( \mathcal{M}(A(S_k)) \) of the arrival time distribution of \( A(S_k) \) (see Bloomfield and Steiger 1983, p. 110f).
4 Solution Approach

In this section, we discuss how we estimate the arrival times and thus the TWs for each request. Because of the spatial and temporal interaction between requests, an analytical approach is not possible, and we turn to approximation. We propose an approach in which we use offline simulation to derive estimates for values of aggregated states. This method allows us to provide state-dependent estimates of arrival times but limits the computation to an offline training period. By training offline, the method allows for estimates to be reported instantaneously once a customer request has been accepted.

Our method of associating values with aggregated states can be viewed as non-parametric as the mapping from the state to the value need not take any particular functional form and is state dependent. Our choice of a non-parametric approach is based on the results reported by state-of-the-art methods in the dynamic vehicle routing literature, all of which use non-parametric approaches to estimate the value of post-decision states (the cost-to-go) (Goodson et al. 2013, 2016, Klapp et al. to appear, Ulmer et al. 2016, 2015, to appear, Voccia et al. to appear) and the fact that a dynamic vehicle routing problem drives our estimates.

Ideally, we would approximate median arrival times for every state. However, as the state space is essentially infinite, we could not possibly store, let alone, calculate values for each state. Thus, we operate on an aggregated state space. Our solution approach makes two additional assumptions. First, a complication in solving the TWAP arises in trying to approximate medians. As we discuss and demonstrate in §A.1 in the Appendix, empirical estimations of median values often exhibit significant instability when only a few observations are available. Similar issues have been raised with the regard to absolute least difference regression (Ellis 1998, 2000). The obvious solution is to increase the number of simulation runs to increase the number of samples per state. With millions of aggregated states, however, even an offline approach coupled with efficient methods for computing running medians (Mohanty 2003, Perreault and Hébert 2007) or methods that do not require storing all observed values (Jain and Chlamtac 1985) are computationally intractable. Interestingly, our results show that the approximation of mean values does not suffer as greatly from the challenges of small sample sizes, and fortunately, median and mean values are often similar, differing significantly only for skewed distribution with long tails (Von Hippel 2005). Second, because we are estimating
Table 2: Anticipatory Time Window Assignment: Notation

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( d(n_k, \theta_k) )</td>
<td>Travel duration to customer ( n_k ) (without service times)</td>
</tr>
<tr>
<td>( \bar{d}(\theta_k) )</td>
<td>Travel duration for tour ( \theta_k ) (without service times)</td>
</tr>
<tr>
<td>( b_k(\theta_k) )</td>
<td>Free time budget left in ( t_k )</td>
</tr>
<tr>
<td>( M(A(S_k)) )</td>
<td>Median of arrival time distribution for customer ( n_k )</td>
</tr>
<tr>
<td>( E(A(S_k)) )</td>
<td>Expected arrival time for customer ( n_k )</td>
</tr>
<tr>
<td>( A_{\text{realized}}(S_k) )</td>
<td>Realization of arrival time for ( n_k )</td>
</tr>
<tr>
<td>( A_{\theta_k}(n_k) )</td>
<td>Expected arrival time at ( n_k ) given ( \theta_k ) (inclusive service times)</td>
</tr>
</tbody>
</table>

means rather than medians, we base our approximations on mean deterministic travel and service times. The mean value is invariant to random service and travel times. To maintain the flow of the paper, we validate both assumptions in detail in §A.1 of the Appendix.

We call our approach the anticipatory time window assignment approach (ATW) with decisions \( X^{\text{ATW}} \). The remainder of this section first presents our state space aggregation and then the offline simulation that we use to find values for the aggregated states. We conclude by describing the approaches that we use as benchmarks for the proposed approach and the tuning of ATW.

### 4.1 State Space Aggregation

A suitable aggregation \( \mathcal{A} \) needs to meet two criteria. First, \( \mathcal{A} \) must reduce the number of states significantly enough to allow for frequent observations of each aggregated state and a sufficient value approximation. Yet, at the same, \( \mathcal{A} \) must allow for the differentiation between states that have different values (Barto 1998, p. 193).

In the following, we motivate and define our aggregation scheme \( \mathcal{A} \). The required notation is depicted in Table 2. The aggregation is motivated by Ulmer et al. (to appear) showing that an aggregation \( \mathcal{A} \) based on temporal state attributes is often effective for the estimation of post-decision state values in dynamic vehicle routing problems. In the case of vehicle routing problems, there is always the question of the effectiveness of relying on only temporal and not geographic information when estimating future costs or in this case arrival times. As we discuss subsequently, our temporal aggregation scheme does implicitly capture information about the geographical distribution of customers in the route. Yet, Ulmer et al. (2016) demonstrate that coupling temporal and explicit geographic information can lead to improved outcomes. However, Ulmer et al. (2016) also relies
on online computation to incorporate the geographic method. As a result, the method proposed in Ulmer et al. (2016) cannot return solutions instantaneously as can our proposed approach. In fact, we are not aware of any successful offline approximation methods for dynamic vehicle routing that do incorporate explicit geographic information.

Because we are seeking to estimate the mean arrival time rather than the value of a post-decision state, however, we propose a different set of temporal parameters than that found in Ulmer et al. (to appear). For ATW, we propose to represent a state $S_k$ by:

- the travel duration, excluding service times, to the new customer $\bar{d}(n_k, \theta_k)$ as determined by $\theta_k$,
- the overall travel duration $\bar{d}(\theta_k)$, again excluding service times, associated with the current partial route $\theta_k$, and
- the amount of time $b_k(\theta_k)$ that is available to integrate new requests after traveling to and serving all of the customers currently included in $\theta_k$. We call the time $b_k(\theta_k)$ the free time budget.

All three measures are specific to the tour $\theta_k$ given by the state $S_k$, without regard for future customers. The aggregation of state $S_k$ is $\mathcal{A}(S_k) = (\bar{d}(n_k, \theta_k), \bar{d}(\theta_k), b_k(\theta_k))$.

The travel duration $\bar{d}(\theta_k)$ is the sum of travel times and neglects the service times. We compute $\bar{d}(\theta_k)$ as

$$\sum_{i=0}^{\lvert \theta_k \rvert} d(\theta_k^i, \theta_k^{i+1}).$$

The value $\bar{d}(n_k, \theta_k)$ is defined as the duration of the subtour of $\theta_k$ ending in $n_k$. Let $n_k$ be the $m^{th}$ stop on $\theta_k$, and $\bar{d}(n_k, \theta_k)$ is computed as

$$\sum_{i=0}^{m-1} d(\theta_k^i, \theta_k^{i+1}).$$

The free time budget $b_k(\theta_k)$ is the difference between the arrival time to the depot $A_{\theta_k}(D)$ if
following $\theta_k$ and the time limit $t_{s}^{*}$.$max$. We compute $A_{\theta_k}(D)$ as

$$
\sum_{i=0}^{\left|\theta_k\right|} \left( d(\theta_k^i, \theta_k^{i+1}) + \zeta \right).
$$

(4)

Then, we compute $b_k(\theta_k)$ as

$$
b_k(\theta_k) = t_{s}^{*} - A_{\theta_k}(D).
$$

(5)

The choice of these aggregation parameters reflects a desire to capture both implicitly and explicitly the number of new requests that might be inserted in the tour in front of the request at the $k^{th}$ decision point while at the same time limiting the dimensions of the aggregated states. The case for the free time budget is straightforward. The lower the free time budget is, the fewer customers will be inserted into the tour in general and before the current request in particularly.

Then, the impact of new customers added to the tour on the arrival time at a customer also depends on the customer’s location and the locations of the other customers. We implicitly capture this information through the remaining temporal parameters and their relationship to the free time budget. Likewise, the relationship between the travel duration to the request inserted at the $k^{th}$ decision point and the total tour duration helps to understand where in the tour the new request is and potentially its geographic proximity. For instance, a short duration to the newly inserted request coupled with a relatively long tour duration tells us that the new request is relatively early in the tour and likely close to the depot. Analogously, a long duration to the request relative to the tour length suggests that the customer is deep in the tour. If the travel duration is short and the free time budget low, we implicitly know that the new request is close to the depot and that few customers may be inserted before this customer.

### 4.2 Approximating the Mean Values

To approximate the mean values for the aggregated states, ATW calculates the running averages of the (aggregated) state observations over 100,000 approximation runs. For each run, a sample of customer requests for a day is generated. Customers are assigned in order of the request based on the VRPSR assignment and routing policy. The state associated with the assigned customer request
is stored. Once we have examined all of the sampled requests for a day, we compute the arrival time for each state and update the running average of the arrival time associated with that state.

Because the number of aggregated states is still vast, we partition the aggregated state space. The best partitioning of the states is not known a priori. We apply a dynamic state space partitioning scheme, the dynamic lookup table (DLT) introduced by Ulmer et al. (to appear). The DLT groups states into partitions and calculates the values for each partition. Over the approximation process, the DLT subsequently updates some partitions by splitting them into a set of new partitions. The division of partitions is initiated in two cases. First, a division of partition $p$ is initiated if the number of observations $O(p)$ of a partition is sufficient. In such a case, a more detailed approximation of smaller partitions is possible. The second mechanism for initiating a division occurs when a partition exhibits significant variance and thus an unreliable estimate indicated by the standard deviation $\sigma(p)$. We draw on the product of both parameters relative to the average parameters over all partitions $\bar{O}$, $\bar{\sigma}$ as an indicator. This product is then compared to a threshold parameter $\tau$ as

$$\frac{O(p) \times \sigma(p)}{\bar{O} \times \bar{\sigma}}.$$  (6)

If the threshold is exceeded, the division is induced by splitting the intervals in all three dimensions. This results in $2^3 = 8$ new partitions. The DLT draws on an initial, equidistant partitioning with intervals of 16 minutes in every dimension and decreasing interval lengths of 16, 8, 4, 2, and 1. The disaggregation threshold is set to $\tau = 3.0$. It is possible that not all of the aggregated states will be visited during the approximation phase. In the few cases in which this happens, ATW draws on the approximation of benchmark $X^{\text{linear}}$ defined in the following section.

### 4.3 Benchmark Heuristics

In this section, we present two benchmark approaches to which we will compare our ATW. The main benchmark, the linear approximation $X^{\text{linear}}$, is the analytical equivalent to ATW in that it operates only on temporal parameters. The linear benchmark also mimics the approach presented in Madsen et al. (1996). The second benchmark is a naive myopic approach.

The idea of $X^{\text{linear}}$ is to equally distribute the free time budget $b_k(\theta_k)$ over the tour with respect to spatial distribution of customers. The benchmark $X^{\text{linear}}$ focuses on only the scheduling aspect of
the TWAP and the underlying vehicle routing problem. Particularly, for a given state $S_k$, $X^{\text{linear}}(S_k)$
distributes the free time budget $b_k(\theta_k)$ with respect to the percentage of $\bar{d}(n_k, \theta_k)$ to $\bar{d}(\theta_k)$. This
distribution of free time budget can be viewed as a buffer against shifts caused by the addition of future requests to the route. Specifically, we compute $X^{\text{linear}}(S_k)$ as

$$X^{\text{linear}}(S_k) = A_{\theta_k}(n_k) + \frac{\bar{d}(n_k, \theta_k)}{d(\theta_k)} \times b_k(\theta_k).$$

(7)

Notably, for cases with short service times, the free time budget may not be consumed for every realization, and the linear distribution may lead to a systematic overestimation. To avoid this overestimation, we reduce the time budget with respect to the average arrival time $A_{\text{average}}(D)$ at the depot for each instance setting as depicted in Equation 8:

$$X^{\text{linear}}(S_k) = A_{\theta_k}(n_k) + \frac{\bar{d}(n_k, \theta_k)}{d(\theta_k)} \times \max(b_k - T + A_{\text{average}}(D), 0).$$

(8)

We add the non-negativity in the last term to avoid a negative shift in cases when $A_{\theta_k}(D) > A_{\text{average}}(D)$. For the computational study, $A_{\text{average}}(D)$ is approximated by 10,000 test runs.

Beside $X^{\text{linear}}$, we apply a myopic approach $X^{\text{myopic}}$ to analyze the impact and highlight the necessity of anticipation. This approach ignores future developments and estimates the arrival time given the current state $S_k$ and notably $\theta_k$: $X^{\text{myopic}}(S_k) = A_{\theta_k}(n_k)$.

5 Experimental Design

In this section, we describe the instances parameters and discuss our implementation. The instance settings are the combination of the customer distribution layout, the application represented by service time and requests rate, and of the decision policy of the underlying VRPSR. As demonstrated in §A.1, stochastic travel and service times have minimal impact on the solution values. Importantly, given that means are most likely to misestimate medians when the tail of the distribution is “fat,” the results in §A.1 show that, even in the presence of the relatively “fat” tails found in the most congested metropolitan areas, our use of the mean is an effective estimate of the median. As a result, in our computational experiments, we assume deterministic travel and service times.
5.1 Geography

The depot and the customer locations are derived from a database of residents in the Iowa City area, Iowa, USA. The area has a population of around 160,000. We use the residencies as potential customers. Overall, there are 32,850 locations. The layout of the area of Iowa City and a sample of customer locations are shown in Figure 3. We cluster the data into three clusters, and these clusters are determined by a $k$-means clustering algorithm. The east side of Iowa City, represented by triangles, includes the downtown and the campus of the University of Iowa, and due to the number of apartment buildings particularly serving the students of the University, this area is densely populated. The west side of Iowa City, indicated by the squares, is less densely populated and more heterogeneously distributed, partly due to the large amount of land consumed by the University’s hospital, football stadium, and golf course. Directly adjacent to Iowa City proper is the city of Coralville, indicated by the circles.

We denote the customer distribution as $C_1$ for Coralville, $C_2$ for east side of Iowa City, and $C_3$ for the west side of Iowa City. We select the depot location $D$ in the north-west close to the highway crossing indicated by the $+$ symbol. This location is suitable, e.g., for less-than truckload or warehouses, but also for many workshops usually located close to Coralville. In our computational evaluation, we examine four cases to analyze how our proposed approach and the benchmarks perform for the different customer distributions. First, we consider the case in which a vehicle serves only one of the three clusters. We denote these as $C_1$, $C_2$, and $C_3$, respectively. Second, we assume that a vehicle serves customers in any of the clusters. We denote this distribution as $C_{all}$.

We calculate the distances between two customers by the Haversine distance measure (Shumaker and Sinnott 1984). This distance measure is the equivalent to the Euclidean distances on a globe. To account for the road network, we multiply the Euclidean distances by a factor of 1.4, the multiplier which Boscoe et al. (2012) shows closely relates Euclidean with actual driving distances in most US cities. The vehicles travel with 30km per hour. Travel durations are rounded up to minutes, and the minimal travel duration between two customers is 1 minute.
Figure 3: Depot Location and Customer Distributions for the Iowa City Area
5.2 Service Time and Request Rate

We assume that customer requests are made the day before service is provided. This leads to a capture and an execution phase, each of which we assume to be $t_{\text{max}}^c = t_{\text{max}}^s = 480$ minutes. We analyze the suitability of ATW for different applications mainly differing in the time required to serve the customer. The service time of the instances is selected as $\zeta \in \{2, 5, 15, 30\}$. Time $\zeta = 2$ represents an application where drivers visit only customers and directly proceed with their tour, e.g., for parcel pickup or delivery. Time $\zeta = 5$ represents applications where loading may require more time, e.g., for grocery delivery or where an actual short service is conducted such as reading a meter. For more complex services, we select service time of $\zeta = 15$ suitable for a routine maintenance and short technical tasks, e.g., drain cleaning or the activation of TV- or internet-cable. Finally, we set $\zeta = 30$ to represent longer and more complex technical tasks, e.g., repairing appliances.

To simulate different requests rates, we vary the expected number of requests $c = 20, 30, 50, 80, 100$. We generate requests over the time horizon of the capture phase $T^c = [0, t_{\text{max}}^c]$ via a Geometric distribution assuming discrete minute-by-minute time steps.

The combination of the instance setting’s dimensions theoretically leads to an overall set of 80 different instance settings. However, because of the service times, the number of customers who can actually be served is limited. For example, consider the case of a service time $\zeta = 30$. Ignoring travel time, at most $\frac{480}{30} = 16$ customers can be served given this travel time. In fact, when $\zeta = 30$, the results are therefore the same for all $c$. The same is true for $\zeta = 15$ and $c = 50, c = 80, c = 100$. Hence, we remove the instance settings $\zeta = 30, c > 20$ and $\zeta = 15, c > 50$. This leads to overall 64 instance settings.

5.3 Assignment and Insertion Policy

For the VRPSR, decisions are made about the assignment of a request to a vehicle and the routing of the customers. Like Campbell and Savelsbergh (2005), we route customers using Cheapest Insertion. Cheapest Insertion builds the tour by inserting new requests into an existing tour such that the increase in tour duration $\bar{d}(\theta_k)$ is minimum (Rosenkrantz et al. 1974). Our acceptance decisions are non-selective, meaning that each feasible request is accepted and inserted into the tour, a procedure usually applied in practice. For the detailed algorithm, we refer the reader to §A.3 in
the Appendix. By comparing the tours resulting from cheapest insertion with an optimal tour, we demonstrate in § A.1.3 that this choice has minimal affect on the total route duration.

6 Computational Results

In this section, we evaluate the solution quality of the proposed solution approach. We first define a set of measures and compare the proposed approach to the benchmarks with respect to the measures and instances’ parameters. We then analyze the differences in solution quality for customers who request early in the day versus those requesting later.

6.1 Measures

In the following, we define the measures that we use to analyze the results. The objective of the TWAP is to minimize the absolute difference of estimated arrival times by policy $X$ and realized arrival times per customer. We denote the average of these differences with $Q(X)$. To determine customer satisfaction, we further need to analyze the average maximum difference per day and vehicle between estimated and realized arrival time $Q_{\text{max}}(X)$. Finally, we analyze the results with respect to the percentage of customers served within a TW of $\nu$ minutes, denoted by $Q_{\text{TW}}^{\nu}(X)$ with $\nu = 30, 60, 120$ minutes. We assume that the TWs are centered at the estimated arrival time. As an example, $Q_{\text{TW}}^{120}(X)$ measures the percentage of customers with difference of less than plus or minus 60 minutes from the estimated arrival time.

6.2 Comparison Against the Benchmarks

In this section, we compare the proposed ATW approach to the benchmarks described in §4.3. For each instance setting, we run 100,000 test runs. First, we present the results averaged over all 64 instance settings. Then, we analyze the results with respect to the instance parameters.

Table 3 presents the average results. For the ATW, the linear, and the myopic approaches, the rows of the table represent the average difference from the estimated arrival time, the maximum difference, the percentage of customers served within a 30-minute TW, the percentage of customers served within a 60-minute TW, and the percentage of customers served within a 120-minute TW,
respectively. On average, for the solution returned by the ATW, the estimated arrival time and the realized arrival times differ by 19.1 minutes. The average difference for $X^{\text{linear}}$ is less than a half an hour, and for $X^{\text{myopic}}$, it exceeds 80 minutes. The results for the myopic approach demonstrate the value of anticipating future requests when estimating arrival times. The maximum difference for $X^{\text{ATW}}$ is 74.1, more than 30 minutes less than $X^{\text{linear}}$.

Looking at the percentage of customers served within a TW, we can see that 58.8% of the customers are served within a 30-minute TW centered at the estimated arrival time given by the ATW method. Further, for one- and two-hour TWs common in many practical applications, $X^{\text{ATW}}$ meets on average $Q^{60\text{TW}}(X^{\text{ATW}}) = 80.3\%$ and $Q^{120\text{TW}}(X^{\text{ATW}}) = 94.6\%$ customers within the TWs. Neither the linear nor myopic approaches are capable of such performance. However, the linear approach does close the gap as the TW widens. This result suggests that the simple linear approach may be amenable in cases in which a provider offers wide TWs.

Figure 4 shows the disaggregation of the results from Table 3. Each panel of the figure shows the average difference from the estimated arrival times, with respect to service time, expected number of customers, customer distribution, and solution approach. On the global y-axis, the service time is depicted. On the global x-axis, the number of customers is shown. Each entry of the global figure therefore represents the results for a specific service time and expected request combination. These results are then differentiated by the solution method on the y-axis and the customer distribution indicated by the grayscale of the bars.

We observe a general pattern. The difference increases with the expected number of customers and the service time. For $\zeta = 2, c = 20$, depicted in the upper left of Figure 4, the average difference $Q(X^{\text{ATW}})$ is less than 10 minutes and even for the myopic approach less than 30 minutes. An increase in expected number of customers and service time leads to higher differences. In the worst case with $\zeta = 30, c = 20$, the difference reaches half an hour for ATW, the amount of one service

### Table 3: Solution Quality

<table>
<thead>
<tr>
<th>Measure</th>
<th>$X^{\text{ATW}}$</th>
<th>$X^{\text{linear}}$</th>
<th>$X^{\text{myopic}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Q$</td>
<td>19.1</td>
<td>26.3</td>
<td>81.0</td>
</tr>
<tr>
<td>$Q_{\text{max}}$</td>
<td>74.1</td>
<td>105.0</td>
<td>261.5</td>
</tr>
<tr>
<td>$Q_{30\text{TW}}$</td>
<td>58.8</td>
<td>48.5</td>
<td>17.9</td>
</tr>
<tr>
<td>$Q_{60\text{TW}}$</td>
<td>80.3</td>
<td>70.5</td>
<td>32.7</td>
</tr>
<tr>
<td>$Q_{120\text{TW}}$</td>
<td>94.6</td>
<td>88.9</td>
<td>53.6</td>
</tr>
</tbody>
</table>
Figure 4: Average difference in minutes between the estimated arrival times and actual arrival times with respect to service time, expected number of customers, customer distribution, and solution approach.

time, and even two hours for the myopic approach.

The development can be explained by the individual impact of both the service time and the number of customers. The greater the number of customers, the more likely a customer is to be shifted in the route. That is, for a given customer request, the arrival time distribution is more variable as more customer requests are expected. Further, if the service time is high, every newly assigned customer results in a significant shift of subsequent existing customers in the tour. Again, the range in the arrival time distribution is high, making accurate prediction more difficult. Still, ATW is able to anticipate the shifts induced by the new customer requests, particularly, for a high number of requests and lower service times. Given low service times of $\zeta = 2$, the average difference does not exceed 18.4 minutes, while for the highest service time $\zeta = 30$, the difference is always higher than 26.8 minutes. This result suggests that applications with lower service times, such as home attended delivery, may be more more suited for narrow TWs compared to applications with high service times such as complex repair-services.

A similar pattern is observed in the case of TWs. Using a similar layout as Figure 4, Figure 5
the percentage of one-hour TWs met by the ATW. Again, the service quality is high for low service times and fewer numbers of customers. Service quality then decreases with increasing service time and number of customers.

6.3 The Value of Making a Request Late in the Day

There are also patterns depending on when during the day a customer calls. Particularly, arrival times for early requests are difficult to approximate while later requests allow a more accurate prediction. As an example, we present results for \( c = 50, \zeta = 15, C_{all} \) because the results are slightly below average with \( \mathcal{Q}_{\text{TW}}^{90} = 76.4\% \) and \( \mathcal{Q} = 21.4 \) minutes. We analyze the differences between the realized arrival time and the estimated arrival time of ATW \( \gamma^+(X) = \max\{A - X, 0\} \) and \( \gamma^-(X) = \min\{A - X, 0\} \). The case \( \gamma^+(X) > 0 \) indicates a late arrival, \( \gamma^-(X) < 0 \) indicates an early arrival.

Figure 6 depicts the average difference of the realized arrival times from the estimated arrival times with respect to the point of time \( t \in [0, t_{c_{\text{max}}}^c] \) in the capture phase. The results are averaged over all observed states \( S_k \) with \( t_k = t \). For the purpose of presentation, we only plot values for
Figure 6: Difference between estimated and realized arrival times by time of day of the request for 50 expected customers, a service time of 15 minutes, and all customers

points of time with more than 10 observations. On the x-axis, we plot time. On the y-axis, we plot the positive differences $\gamma^+$ and the negative divergences $\gamma^-$. Generally, because of the constraint on the length of the technician’s working day, not every request can be assigned by the VRPSR and observations for late requests with $t_k \geq 300$ are rare. Thus, customers calling at the end of the capture phase may not be served the following day. On average, the vehicle is 63 minutes late or 38 minutes early. Yet, Figure 6 exhibits decreasing values for $\gamma^-$ and $\gamma^+$ over time. Early requests experience a greater difference between current and realized arrival time. Customers requesting service early in the horizon of the capture phase encounter a tour that is not yet established and to which many requests are still to be assigned. Over time, the tour becomes more established and there are fewer opportunities for customers to be inserted. Thus, estimates of the arrival time improve. After 3 hours, the average difference is 10 minutes early and 12 minutes late. This decrease continues until the difference is near zero around $t = 300$.

These results have important implications for companies seeking to offer time windows to customers. Customers prefer shorter waiting times, such as is given by a narrower time window. However, when guaranteed a waiting time, customers’ dissatisfaction increases if the time guarantee is not met (Kumar et al. 1997). Thus, companies must balance customers’ desires for short waits with the impact on customer satisfaction for failing to meet those waits. Our results suggest that
state-dependent TWs may be one way to achieve greater satisfaction. Generally, one strategy might be to offer wider time windows to customers requesting earlier in the capture phase and narrower time windows to those requesting later. To analyze the potential of state-dependent TWs, we consider the time of the requests and naively set the TW-sizes for the instance setting to $\nu = 120$ minutes for customers requesting in the first two hours $t_k \leq 120$ and $\nu = 60$ minutes for all later requests. For the realizations, two-hour TWs are communicated to 55.6% and one-hour TWs to 44.4% of the assigned customers. In 89.2% the customers are served within the TWs. The result is slightly worse than only looking on two-hour TWs with 93.0% and significantly better than one-hour TWs with 76.4%. So, heterogeneous TW-layouts over time may allow both reliability and accuracy and overall higher customer satisfaction.

7 Conclusion

In this paper, we introduce a method for estimating state-dependent arrival times and thus time windows for customers requesting service over the course of the day. The time windows must be communicated at the time of the request, before all requests for the day are known. While it is optimal to estimate the median arrival time for customers, we propose an approach that seeks to estimate mean arrival times. We do this to overcome the instability that arises when estimating medians. To estimate the mean values, we propose an offline simulation approach that seeks to provide state-dependent estimates. However, because of the large state space, we operate on aggregated states. We propose an aggregation scheme based on the temporal attributes of the state and partition the state space using an approach that responds to the data gathered during the estimation procedure. Our results show that the estimates of the mean are more stable and a better proxy for the true median than our attempt to approximate the median values. Further, our results show that the proposed approach significantly outperforms two benchmarks. The comparisons against the benchmarks also indicate that the estimates benefit from state-dependent information and from explicitly accounting for future requests.

The results also offer two suggestions for companies seeking to balance time window widths and on-time performance to improve customer satisfaction. First, the most accurate predictions of arrival times are in cases with lower mean service times. These are the companies who can offer the
most narrow time windows. Thus, companies should use time window schemes that reflect their industry characteristics. Second, because of the uncertainty associated with the future request, both in terms of number but also their geographical dispersion, customers who request earlier in the time horizon might be most satisfied if offered a wider time window. As the day progresses and more is known about the route that will be executed, requesting customers can be given increasingly narrower time windows.

There are many avenues for future work related to the TWAP. First, the work in this paper focused on the use of the mean arrival time as an approximation for the median, the optimal value when minimizing absolute deviations. However, the mean is itself the optimizer for objectives related to the squared deviations. It would be worthwhile to explore the performance of the proposed approach in cases in which the objective was to minimize squared deviation. Further, given our results that demonstrate the potential value of state-dependent time windows, it would be valuable to extend this work to optimize the time-window width for each state. In fact, future research might consider optimizing state-dependent time windows directly rather than fitting time windows to optimized arrival times, the heuristic used in this work. Such a problem would pose significant challenges as one needs to both approximate the arrival time distribution as well as optimize the width and starting time of each state-dependent time window. In addition, future research could consider an approach that integrated customer acceptance, routing, and time-window determination.

More generally, we believe that an approach such as is proposed here could be used to price time slots in attended home delivery. As indicated in §2, the existing approaches are all online approaches and may be sacrificing solution quality for real-time computation. Our offline approach can overcome that challenge. Finally, for this problem and many problems in dynamic vehicle routing, there could be value in developing an offline approximation approach that incorporates geographic information.

Acknowledgment

We thank Todd Kruse for his help in attaining the database of residents for Iowa City, Iowa, and Coralville, Iowa.
References


Appendix

In the Appendix, we present a validation of our assumption of model and approach, the MDP-model for the VRPSR, the proof that the mean values are invariant to random travel and service times, and finally the detailed results of the computational study.

A.1 Validation

Our solution approach makes a number of assumptions relative to the model. In this section, we explore the impact of our modeling choices and solution approach. We first examine the effect of solving for the mean versus the median. Our results show that we get better and more reliable results by approximating the mean. Further, we discuss the impact of modeling travel and service times as deterministic values as well as the quality of Cheapest Insertion routing.

A.1.1 Approximating the Mean versus the Median Value

In this section, we demonstrate that our approximation of the mean of the arrival time distribution is a suitable proxy for the median in this problem. To this end, we describe how we estimate the median for a state. We then compare the solution quality based on the median approximation to ATW and provide reasons for the weak median approximation. Finally, we show that the difference in median and mean value is low.

**Median Approximation.** To approximate the median, we draw on the same procedure of aggregation and simulation as ATW described in §4. Still, we additionally require an estimate of the arrival time distribution for every (aggregated) state. Hence, we store the number of observations for every realized arrival time for every state. After the approximation phase, these empirical distributions are then used to calculate the median for every aggregated state. Technically, we integrate the realized arrival $A_{\text{realized}}(S_k)$ time as an additional dimension of our aggregation, now leading to a 4-dimensional vector $\mathcal{A}^M(S_k) = (\bar{d}(n_k, \theta_k), \bar{d}(\theta_k), b_k(\theta_k), A_{\text{realized}}(S_k))$. As ATW, we run 100,000 approximation runs and apply (in this case a 4-dimensional) DLT with identical settings as ATW. Within the simulations, we then count the number of observations for each particular state $\mathcal{A}^M(S_k)$ and use these values to calculate the empirical median for each 3-dimensional state $\mathcal{A}(S_k)$.
Comparison to ATW. We now look at the performance of the estimated mean versus median values. Figure A2 shows the average difference for \( C_{\text{all}} \), for each service time value and each number of customers for both estimated means and medians. As discussed in §5, the figure does not report values for \( \zeta = 30, c > 20 \) and \( \zeta = 15, c > 50 \). The results show the estimations of the medians leads to significantly larger differences from the estimated times than do the estimates of the means. Similarly, Figure A1 shows the percentage of one-hour TWs centered at the median and mean values, respectively, that would be met given the estimates of the arrival time. As with Figure A2, the results are given for each service time and number of customers. The results show that the estimated mean values lead to significantly better performance with respect to the TWs.

Approximation Performance. To better understand the poor performance of the estimated medians relative to the estimated means, we look more closely at the estimates. To this end, we need to analyze the structure of the arrival time distribution. Figure A3 shows the empirical arrival time distribution at a customer \( n_k \) over \( n = 10,000 \) realizations for a randomly selected state with time \( t_k = 87, \mathcal{A}(S_k) = (\bar{d}(n_k, \theta_k) = 44, \bar{d}(\theta_k) = 97, b_k = 263) \) and \( A_{\theta_k}(S_k) = 87 \) drawn from
Figure A2: Percentage of one-hour Time Windows Met for Estimated Median versus Mean

Figure A3: Arrival Time Distribution for State $S_k$
Figure A4: Development of $M(A(S_k))$ and $E(A(S_k))$ for State $S_k$

the instance setting with service time $\zeta = 15$ minutes, expected number of requests $c = 50$, and
customer distribution $C_{all}$. On the x-axis, the realized arrival times $A_{realized}(S_k)$ are depicted. On
the y-axis, the percentage of realizations for each arrival time is shown. We can observe a set of
distinguished peaks combined with some noise. The difference in time between the peaks is between
16 and 20 minutes. Since the service time is 15 minutes, the shift between two peaks results from
the integration of a new request in $\theta_k$ before customer $n_k$ under consideration. Between the peaks,
the number of observations is low and, in an extreme case, can be associated with singularities
in the distribution meaning that in these areas single observations may significantly impact the
estimation of the median (Ellis 2000). As we show in the following, this structure of the arrival
time distribution indeed impedes the empirical calculation of the median. We further show that the
mean $E(A(S_k))$ is a suitable proxy for estimating $M(A(S_k))$ for this state.

Figure A4 shows the development of both empirical median $M(A(S_k))$ and mean value
$E(A(S_k))$ for the aforementioned state. On the x-axis, the number of observations is depicted.
On the y-axis, the current empirical median and mean are shown. The dashed line indicates the
empirical median over all 10,000 observations. We observe that the smoothing by the mean value
subsequently allows convergence while the median shows many jumps when the number of ob-
servations is small. After 20 observations, the mean value only differs by about seven minutes to
Table A1: Percentage of Early and Late Arrivals

<table>
<thead>
<tr>
<th>Distribution</th>
<th>early</th>
<th>late</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_1$</td>
<td>52.4</td>
<td>47.2</td>
</tr>
<tr>
<td>$C_2$</td>
<td>52.2</td>
<td>47.3</td>
</tr>
<tr>
<td>$C_3$</td>
<td>52.4</td>
<td>47.2</td>
</tr>
<tr>
<td>$C_{all}$</td>
<td>51.9</td>
<td>47.6</td>
</tr>
</tbody>
</table>

the empirical median after 10,000 observations while the current median is stuck in the “wrong peak” and shows a difference of 20 minutes. The median converges to the mean value after 50 observations. This behavior exhibits a similar behavior to that described in Ellis (1998) and Ellis (2000). Notably, for some distributions, the least average difference method shows high instability when only a few observations are available. The results in Figure A4 show such behavior. In general, in our offline simulations, states are observed with heterogeneous frequency and often less than 10 times. As a result, a reliable calculation of the median for every state is not possible.

**Median vs. Mean.** While our estimated medians perform poorly relative to the estimated mean, the results do not say anything about how well the estimated means estimate the true medians. In the example given in Figure A4, the mean value after 10,000 runs is 248.2 and only slightly different than the median of 248. To understand the general relationship between the estimated means and the true medians, we analyze the percentage of early and late arrivals based on the estimated arrival time given by the ATW. If the percentages are identical, the median and the mean value are the same. As an example, we consider $c = 50$, and $\zeta = 15$. Table A1 shows the percentages of early and late arrivals for each the four customer distributions. The sums of each row do not add to 100% because some of the observations have a difference of zero. The results show that vehicles arrive in slightly more cases too early than too late. This means that the median is in some cases lower than the mean value. Thus, despite being a better proxy for the true median than the estimated median values, ATW’s estimates of the mean on average slightly overestimate the median. However, the ATW approach based on mean values significantly outperforms ATW based on median estimations.

In the following, we analyze how over- and underestimates of the arrival times depend on the parameters of the state. Preliminary tests reveal that the difference is not significant for $t_{k}$, and the according dependent parameters $b_{k}$ and $\bar{d}(\theta_{k})$. Mainly, a difference with respect to varying $\bar{d}(n_{k}, \theta_{l})$
can be observed. Figure A5 depicts the percentage of early and late arrivals for states differing the duration $\bar{d}(n_k, \theta_k)$, i.e.,

\[
\frac{|\{S_k : \bar{d}(N_k, \theta_k) = d, \gamma^+(X(S_k)) > 0\}|}{|\{S_k : \bar{d}(n_k, \theta_k) = d\}|} \tag{A1}
\]

Generally, the difference in marginal. We can observe a slight shift between to the percentage of too early and too late arrivals. For durations lower than 30 minutes, the percentage of late arrivals is marginally higher than the percentage of early arrivals. For $30 \leq \bar{d}(n_k) < 60$, both percentages are identical, and for $60 \leq \bar{d}(n_k) \leq 120$, the percentage of early arrivals is higher explaining the overall slightly higher percentage of early arrivals shown in Table A1. A higher percentage of early arrivals indicates that the median is lower than the mean and vice versa. Hence, we can assume varying arrival time distributions with respect to the travel duration to a customer.

A.1.2 Random Travel and Service Times

As discussed in §4, we use mean service and travel times in our estimates of the mean arrival time and in our computational evaluation. It is straightforward to show that doing so does not impact our estimate expected arrival times. We provide a proof in §A.1.2. Still, stochastic travel and service
times could affect our ability to use the mean as a proxy for the median. Notably, while using mean travel and service times does not affect our estimate of the mean arrival time, it does so for medians only in the case of certain distributions, e.g. the Normal. In this section, we first present a proof demonstrating that using the mean travel and service times does not affect our estimate of the mean arrival time. We then show that, even when variation is high in both the travel and service times, the solution values are relatively unaffected.

**Proof that Estimate is Invariant to Random Travel and Service Times**

In the offline approximation, we set all travel and service times to mean values. In the following result, we demonstrate that doing so does not theoretically affect our estimate of the means.

To show that the approximated arrival time is invariant to use of the mean travel and service times, assume that we are in state $S_k$. We let $\hat{d}_m^i$ be the random cost of adding the $i^{th}$ customer before some customer $\theta_m^k$ on the tour $\theta_k$ associated with the state $S_k$. Then, let $Y_i^m = \hat{d}_m^i - \mathbb{E}[\hat{d}_i | S_k]$ be the random difference between the mean and random costs of inserting the $i^{th}$ before the $m^{th}$ customer on the tour $\theta_k$. We note that the number customers inserted before $\theta_m^k$ is random number $N$, but is bounded by a $K_{max}$ due to the service time per customer. Then, we have

$$
\mathbb{E}\left[ \sum_{i=1}^{N} Y_i | S_k \right] = \mathbb{E}\left[ \mathbb{E}\left[ \sum_{i=1}^{n} Y_i | S_k, n \right] \right]
$$

$$
= \sum_{n=0}^{K_{max}} \mathbb{E}\left[ \sum_{i=1}^{n} Y_i | S_k, n \right] P(N = n | S_k)
$$

$$
= \sum_{n=0}^{K_{max}} \left( \sum_{i=1}^{n} \mathbb{E}[Y_i | S_k, n] P(N = n | S_k) \right)
$$

$$
= \sum_{n=0}^{K_{max}} \left( \sum_{i=1}^{n} \mathbb{E}[\hat{d}_i - \mathbb{E}[\hat{d}_i | S_k] | S_k, n] P(N = n | S_k) \right)
$$

$$
= \sum_{n=0}^{K_{max}} \left( \sum_{i=1}^{n} \left( \mathbb{E}[\hat{d}_i | S_k, n] - \mathbb{E}[\mathbb{E}[\hat{d}_i | S_k, n]] \right) P(N = n | S_k) \right)
$$

$$
= 0.
$$

The first equality follows from the Law of total expectations and the second by definition. Equation (A2) results from the linearity of the expectation operator and Equation (A3) by definition. Equation (A4) follows again for the linearity of expectations, and the final equality follows from the
fact that an expectation is a deterministic value.

**The Impact of Random Travel and Service Time on Solution Values**

To analyze the impact of stochasticity, we model $\zeta$ and/or $d(i, j)$ as random variables following a Gamma-distribution. The selection of the distribution is suitable because it allows the modeling of a skew and a tail, often applicable for service and travel times (Polus 1979, Schmid 2012). We note that adding uncertainty in the travel and service times may lead to violations on the assumption of the maximum length of the working day. As noted in §3, we add customers so that the average travel and service times associated with a route do not violate the constraint on the length of the working day. In the execution of the route, however, we allow this constraint to be violated.

To determine the degree of uncertainty, we evaluate the performance of the means estimated with the proposed ATW method and varying the coefficients of variation (CoVs). Given that means are most likely to misestimate medians when the tail of the distribution is “fat,” we sought a range of values that would represent these possible tails. The literature suggests that the CoV in travel time never exceeds the value of 0.6, with values as high as 0.6 being found only at peak times in highly congested areas such as downtown Manhattan (Yazici et al. 2014). For a reference point, Schmid (2012) considers CoVs between 0.4 to 0.6 for service times, Ehmke and Campbell (2014) consider CoV between 0.1 and up to 0.25 in peak hours for travel times.

Figure A6 presents the average difference per customers from the estimated mean and the actual arrival times for CoV values of 0.0, 0.2, 0.4, and 0.6 for $C_{all}$. As the CoV increases, we expect to the performance to decline since the overall uncertainty increases as well. Figure A6 shows this decline is minimal even when using solutions created without considering stochastic travel and service times. The average difference between the results with a CoV of 0.0 and a CoV of 0.6 is only 2.5 minutes. Figure A7 shows the analogous results for the percentage of one-hour time window violations. Again, the performance declines only slightly as the CoV increases. Stochastic travel and service times have minimal effect on the results.

Nonetheless, given our inclusion of service times, we also consider high values of CoV and accordingly vary the CoV between 0.0 and 1.0 in steps of 0.1 for the instances for the class of instances in which $c = 50$, $\zeta = 15$, $C_{all}$. We test uncertainty in service or travel time respectively as well as uncertainty in both capacities. We calculate the increase in average difference compared to CoV of 0.0. A summary of the results is shown in Figure A8. As expected, the difference increases
Figure A6: Difference from Estimated Mean Arrival Teams When Coefficients of Variations Range from 0.2 to 0.6

Figure A7: Percentage of One-Hour Time Windows Met with Coefficients of Variations Ranging from 0.2 to 0.6
with increasing CoV. The increase for travel time is slower since for this instance setting, with 337.1 minutes, the majority of time is spent for services and only 136.4 minutes on average for routing. Nonetheless, even a CoV of 1.0 leads to additional differences of less than five minutes.

A.1.3 Cheapest Insertion Routing

In the simulations, we apply Cheapest Insertion (CI) routing for two reasons. First, while some routing approaches significantly change the previous sequence of customers, CI allows to maintain this sequence and is therefore well suited for arrival time estimation. Second, CI is easily applied without significant calculation effort and, therefore, reflects routing decisions in practice. Still, CI may result in relatively long routes compared to optimal routes. To analyze the improvement potential of the CI routes, we compare the route durations with optimal TSP solution derived by CPLEX. We draw on the aforementioned instance setting with \( c = 50, \zeta = 15, C_{\text{all}} \), providing travel times between customers up to 45 minutes. We select 100 instance realizations and compare the CI-tour through the final set of customers to the optimal TSP tour. The travel duration for CI-routing is 137.1 minutes while 132.0 for the TSP solutions. On average, the optimal solution reduces travel time by 5.1 minutes. Still, since service times stay the same, the overall sums of travel and service times are 474.0 and 468.9 minutes respectively and the percentual reduction is marginal with only
A.2 Markov Decision Process Model for the VRPSR

In this paper, we are seeking to estimate the arrival times for a vehicle for the case in which customers request service prior to route execution but require confirmation and an estimated arrival time at the time of the request. The related vehicle routing problem is the vehicle routing problem with stochastic requests (VRPSR). In the following, we describe the MDP for the underlying VRPSR. Notably, we change the notation to allow a differentiation to the TWAP by adding a hat to the VRPSR’s state and decision denotation. A decision epoch $k$ occurs whenever a customer request service. State $\hat{S}_k = (t_k, N_k, \theta_k, n_k)$ contains the request’s point of time $t_k$, the set of already assigned customers $N_{k-1}$ and their planned tour $\theta_k$, as well as the new customer $n_k$. A decision $\hat{x}$ determines whether the customer is assigned to this vehicle and where it is inserted in the tour. The reward $R(\hat{S}_k, \hat{x}_k)$ of a decision is one, if the customer is assigned and zero else. The decision leads to an update of the tour $\theta_k$ to $\hat{\theta}_k$ and the set of customers $N_k$. This leads to a post decision state $\hat{S}_{\hat{x}}_k = (t_k, N_{\hat{x}}_k, \hat{\theta}_k)$. If the new request is assigned, this post-decision state and $n_k$ are then the foundation for the TWAP-decision with $S_k = \hat{S}_{\hat{x}}_k$. The stochastic transition $\omega_k$ reveals a new request $n_{k+1}$ at time $t_{k+1}$ and the new state $S_{k+1} = (t_{k+1}, N_{\hat{x}}_k, \hat{\theta}_k, n_{k+1})$. The initial (post-decision) state $\hat{S}_0 = (0, \emptyset, (D, D))$ at time 0 only contains a tour from depot to depot. The termination state $S_K$ is reached, when $t_K = t_{max}$. The objective for the VRPSR is to find a decision policy maximizing the expected sum of rewards.

A.3 VRPSR-Policy Algorithm

Algorithm 1 describes the procedure of the VRPSR-policy and when the TWAP is solved given a current tour $\theta$ and a new request $C$. First, the new customer is inserted in the tour with function $CIRouting(\theta, C)$ based on the Cheapest Insertion procedure by Rosenkrantz et al. (1974). In case the resulting tour is feasible, checked by function $Feasible$, the customer is added to the tour and the tour is updated. At that point, the arrival time is estimated by solving the TWAP. The algorithm provides the updated tour, and, in case of acceptance the according estimated arrival time.
Algorithm 1: Acceptance and Routing Procedure for the VRPSR

Input: Request $C$, Tour $\theta$

Output: Updated Tour $\theta$, Estimated Arrival Time $X$

1. $X \leftarrow \text{NaN}$ // Arrival Time Estimation Initialization
2. if $\text{Feasible}(\text{CIRouting}(\theta, C)) == \text{true}$ // If Customer can be Inserted
3. then
4. $\theta \leftarrow \text{CIRouting}(\theta, C)$ // Customer is Accepted and Tour is Updated
5. $X \leftarrow \text{TWAP}(\theta, C)$ // Arrival Time Estimation
6. end
7. return $\theta, X$