WORKING PAPER SERIES

Dynamic Time Window Assignment for Next-Day Service Routing

Rosario Paradiso/Roberto Roberti/Marlin Ulmer

Working Paper No. 13/2023

OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG

FACULTY OF ECONOMICS AND MANAGEMENT

Impressum (§ 5 TMG)

Herausgeber:
Otto-von-Guericke-Universität Magdeburg
Fakultät für Wirtschaftswissenschaft
Der Dekan

Verantwortlich für diese Ausgabe:
Rosario Paradiso, Roberto Roberti, Marlin Ulmer
Otto-von-Guericke-Universität Magdeburg
Fakultät für Wirtschaftswissenschaft
Postfach 4120
39016 Magdeburg
Germany

http://www.fww.ovgu.de/femm

Bezug über den Herausgeber
ISSN 1615-4274

Dynamic Time Window Assignment for Next-Day Service Routing

Rosario Paradiso*
Department of Operations Analytics, Vrije Universiteit Amsterdam
r.paradiso@vu.nl, *Corresponding Author

Roberto Roberti
Department of Information Engineering, Universitá di Padova
roberto.roberti@unipd.it

Marlin Ulmer
Chair of Management Science, Otto-von-Guericke-Universität Magdeburg
marlin.ulmer@ovgu.de

We consider a problem where customers dynamically request next-day home service, e.g., repair or instalment. Unlike attended home delivery, customers cannot select a time window (TW), but the service provider assigns a next-day TW to each new customer if the customer can feasibly be inserted in the service route of the next day without violating the TWs of the existing customers. Otherwise, the customer service is postponed to another day (which is outside of the scope of this work). For fast service and efficient operations, the provider aims to serve many customers the next day. Thus, TWs have to be assigned to keep the flexibility of the fleet for future requests. For such anticipatory assignments, we propose a stochastic lookahead method that samples a set of future request scenarios, solves the corresponding team orienteering problems with TWs, and uses the solutions to evaluate current TW-assignment decisions. For real-time solutions of the TOP, we propose to approximate its optimal solution value with a tight upper bound. The bound is obtained by solving the linear relaxation of a set packing reformulation via column generation. We test our algorithm on Iowa City data and compare it to several benchmark policies. The results show that our method increases customer service significantly and that our relaxation is essential for effective decisions. We further show that our policy does not lead to observable discrimination against inconveniently located customers.

Key words: next-day service, dynamic TW assignment, stochastic lookahead, approximate dynamic programming, column generation

History: Latest Version: November 6, 2023

1. Introduction

Delivery, repair, maintenance, installation, healthcare, you name it. Each of us receives services that require us to be present at our homes. We usually book the service online in advance and the company performs the service in the following days. With a few exceptions, the timing of the service is out of our hands as service operators aim for efficient routing solutions in highly competitive markets. This leaves customers with day-long Time Windows (TWs) for the day of service. If they are lucky, they might get a more detailed update shortly before the service takes place, e.g., one hour before arrival. However, planning daily activities like work, shopping, or travel is hardly possible due to the requirement of short-notice availability (Zara 2015). There are good reasons why companies communicate such long TWs. At the time a customer requests service, the plan for the day of service is not final as not all customers to be served have requested yet. Large TWs allow flexibility concerning such future requests and, consequently, the service of many customers per day. Communicating smaller TWs can obstruct the integration of future requests, lead to fewer services, and consequently less revenue for the company. Moreover, large TWs not only lead to inconveniences for customers and poor company reputations (Ellis 2011) but can also lead to failed services as customers might not be present and significant recourse costs for the company (Voigt et al. 2023). Therefore, communicating reasonable small TWs may be a win-win situation - *if* they do not reduce the number of services per day noticeably.

In this work, we present means to achieve this goal. We model the problem as a sequential decision process, where customers request service for the "next" day over time. Whenever a customer requests, a decision is made about the (one-hour) TW the customer is assigned to. A decision is feasible if there exists a feasible routing that serves all customers within their respective TWs and allows the vehicles to return to the depot within their working hours. The model also contains the option for not offering a TW to a customer (e.g., due to infeasibility). This customer may then be served at a later day (outside of the scope of this work). The goal is to maximize the expected number of successfully assigned TWs per day. To determine flexible TW-decisions, we propose a stochastic lookahead algorithm for approximating the value of a decision in a state. In every state, we sample a set of future request scenarios. We then evaluate each TW-decision by the average number of sampled requests that can be integrated in the future. This evaluation requires solving a team orienteering problem (TOP) with TWs and mandatory customers. Given the customer waits for a response within minutes, and the need for accurate evaluations (bad evaluations may lead to low-quality decisions), we work with the linear relaxation of a reformulation of the problem with an exponential number of columns (solved via column generation) to benefit from a tight bound. Its strong relaxation allows us to evaluate decisions quickly without sacrificing accuracy.

Based on the approximated values, we use the Bellman Equation to select the decision maximizing the sum of immediate and approximated future rewards. As a tie-breaker, we select the TW with the shortest tour duration.

We test our algorithm on a variety of instance settings based on Iowa City data (Ulmer and Thomas 2019). We compare it with several methodological and practical benchmark heuristics and a perfect information upper bound. We derive the following results:

- Our method significantly outperforms the benchmarks throughout the settings.
- In many cases, our method allows for setting narrow TWs without a substantial reduction of service quality compared to solutions without any TWs.
- Reevaluating the routing in every state is valuable compared to insertion-based procedures.
- Statically assigning TWs to regions of the city leads to inflexibility and a substantial reduction in service availability.
- The dual bound obtained via column generation is essential for accurate evaluations. Relying on the relaxation of the “standard” two-index formulation does not result in effective approximations. The same is true for solving the scenarios with heuristics.
- Our method is surprisingly fair. It does not “sacrifice” customers at inconvenient locations (e.g., on the outskirts of the city). This is noticeable, as previous research has shown that Bellman-driven methodology often discriminates against less convenient customers.

Our paper makes contributions with respect to the problem and methodology. We introduce a new and relevant practical problem and model it as a sequential decision process. We present highly effective decision support advantageous for companies and customers. Finally, we derive several managerial insights for the design and operations of TW-assignments in service operations. We present a tailored method as a combination of approximate dynamic programming and mixed integer programming. The first allows the use of the Bellman Equation for explicit anticipation of future developments. The second allows for a fast but detailed capture of the highly complex decision space. To the best of our knowledge, the concept of tight linear relaxations in lookahead methods for dynamic VRPs is new, and, as our experiments confirm, advantageous compared to heuristic methods usually used in the literature. Thus, the general idea of our method may be beneficial for a broader range of dynamic decision problems with complex routing decisions.

The paper is outlined as follows. In Section 2, we present the related literature. We model the problem in Section 3. Our method is presented in Section 4 and evaluated in Section 5. The paper concludes with a summary and outlook in Section 6.

2. Literature

In this section, we present the related literature. Our work dynamically assigns hard TWs to requesting customers for next-day service with the goal of maximizing services. We will first discuss the most relevant works on the TW assignment for dynamic customer requests. Then, we present works on other TW assignment problems and, finally, other related works.

2.1. Most Related Works

In the following, we discuss the most related works where TWs are assigned to dynamically requesting customers.

In Campbell and Savelsbergh (2005), the authors present a home delivery problem in which customers' requests appear dynamically over time. Similarly to our setting, every time a new request appears the service provider must decide to reject or accept it and assign a TW if the request is accepted. Requests appear during a time frame (capture phase) and the accepted ones will be fulfilled at the end of it, i.e., within a fulfillment phase. In contrast to our setting, the authors assume a very limited number of potential customers and that for each possible customer, the probability that they will request a service is known. The authors propose several insertion heuristics based on different criteria to accept or reject requests. The objective is to maximize the expected profit. The most effective method in Campbell and Savelsbergh (2005) includes request probabilities for future customers in the acceptance or rejection decision. We adapt the method as a benchmark for our experimental study. Since for our problem, the number of potential customers is vast, we do not explicitly incorporate request probabilities but instead apply a multiple scenario approach (Bent and Van Hentenryck 2004).

Another similar problem is studied in Ehmke and Campbell (2014). Here, customer requests appear dynamically and communicate two TWs along their request. The provider can decide whether to assign one of the TWs or reject the request. The authors define acceptance mechanisms to decide whether to assign the first or the second proposed TW or reject the requests based on the likelihood of meeting the TW in the presence of uncertain travel times. In contrast to Campbell and Savelsbergh (2005) and our work, potential future requests are not integrated into the decision. Yu et al. (2023) introduce a richer problem than the one studied in Ehmke and Campbell (2014). The authors consider a TW assignment problem with uncertain travel times, service duration, and customer cancellation. They propose a two-stage stochastic program to capture the uncertainties when assigning TWs. They apply their method to a rolling horizon for each new customer. As in Ehmke and Campbell (2014), the arrival of future requests is not included in the optimization.

Ulmer and Thomas (2019) and Ulmer et al. (2023) consider a dynamic problem in which scheduling and routing are exogenous to the service provider and TWs are soft. In Ulmer and Thomas (2019), the objective is to communicate a service time to the customers in such a way the difference between the communicated time and the actual arrival time is minimized. In Ulmer et al. (2023), instead, the objective is to propose service TWs to customers in such a way that the probability that the TWs will not be violated does not exceed a given threshold and the expected TW size is minimized. The authors characterize optimal TW assignments under some conditions and derive managerial insights supported by computational results. In contrast to Ulmer and Thomas (2019) and Ulmer et al. (2023), our work assigns hard TWs and explicitly decides upon vehicle routing.

2.2. Time Windows Assignment

There is an increasing amount of work on TW assignments without dynamically requesting customers. In contrast to our work, decision-making is usually not dynamic.

The work of Spliet and Gabor (2015) introduces a problem integrating TWs assignment decisions in a Vehicle Routing Problem (VRP) for the first time. The set of customers is known, but their demands are uncertain and revealed only after TWs are assigned to customers. The TW to serve a customer is defined as a continuous partition of the planning horizon. The problem is cast as a two-stage stochastic program in which TWs are decided here and now and routes are wait-and-see decisions. The authors use a branch-and-price-and-cut algorithm to solve the problem.

The same problem is studied in Dalmeijer and Spliet (2018). In this work, the authors propose a different exact approach based on a branch-and-cut algorithm and improve on the results of Spliet and Gabor (2015). Later, Dalmeijer and Desaulniers (2021) focus on addressing symmetry in the formulation introduced in Spliet and Gabor (2015), outperforming the algorithm proposed in Dalmeijer and Spliet (2018). Jabali et al. (2015) study another closely related problem. The authors investigate a VRP with self-imposed TWs. Customer locations and demands are known, but travel times are uncertain. The objective is to define routes and assign TWs (soft and continuous) such that routing cost, TW violation, and overtime routing are minimized.

Spliet and Desaulniers (2015) introduce the discrete version of the TWs assignment VRP. In this setting, the decision-maker selects TWs from a given discrete set. Similarly to Spliet and Gabor (2015), the authors propose a branch-and-cut-and-price algorithm to solve the problem.

The continuous and the discrete versions of the TW assignment problem are also studied in Subramanyam et al. (2018). The authors propose an exact branch-and-bound algorithm. The algorithm decomposes the problem into several deterministic VRPs with TWs, each one corresponding to a node in the branch-and-bound tree.

Neves-Moreira et al. (2018) generalize the problem by allowing split deliveries and including multiple products and periods. The authors solve this problem by using a two-phase matheuristic. The algorithm assigns TWs to products in the first phase. In the second phase, it schedules deliveries.

Extensions of the TWs assignment are presented in Spliet et al. (2018) and Flinterman et al. (2022). In Spliet et al. (2018), the authors introduce time-dependent travel times. In Flinterman et al. (2022), customers are split between early customers and late customers. Early customers appear before a given cut-off time and must be assigned to TWs. Late customers appear after the cut-off time and can be served at any time during the fulfillment phase.

Another problem that combines TW assignment and routing is studied in Celik et al. (2023). Here, the TW assignment problem is coupled with a travel salesman problem. Customers are known, but travel times are uncertain and revealed after assigning the TWs. The objective is to assign TWs in such a way that all the customers can be served and the routing cost is minimized. The authors propose a two-step Benders decomposition algorithm combined with scenario clustering techniques to solve the problem. Numerical results show the benefits of using the two-step decomposition on the considered problems.

2.3. Other Related Works

Finally, we present other related works. Our work is also related to attended home delivery (Fleckenstein et al. 2023, Waßmuth et al. 2023). In such settings, it is assumed that the customer either communicates one TW or is presented with a set of TWs to select from. Thus, the TW-decision is not in the control of the service provider. In contrast to our work, in some attended home delivery problems, fleet resources are not a constraint but rather a cost.

Campbell and Savelsbergh (2006) extend the problem studied in Campbell and Savelsbergh (2005) by introducing customers behaviour and incentives. When a request is accepted, the customer must select a TW for delivery. The customers' behavior is uncertain, but the vendor can alter the probability that a certain TW is selected by offering monetary incentives. The computational results highlight the benefits of incentives on total profit. Customer choices are also studied in Köhler et al. (2023b). The authors use a nested logit model to describe the dependencies between customer choices and TW characteristics.

Sungur et al. (2010) study a multi-period problem. Customer requests appear at the beginning of each day and are characterized by a given and known TW but uncertain service time. The problem assumes that all the possible customer locations are known, but it is not known which ones will appear on a certain day. The objective is to find an a priori master plan that can be easily adapted to daily requests. The objective is to define daily routes that are as similar as possible to the master plan according to some similarity measures. The authors model the problem as a two-stage stochastic program and solve it with a heuristic insertion algorithm. We adapt the master plan of Sungur et al. (2010) as a benchmark for our work where an a-priori tour is found through areas of the city, and, according to the tour, TWs are assigned for each of the areas.

Köhler et al. (2023a) propose a data-driven approach for the problem considered in Campbell and Savelsbergh (2005). The idea is to use historical data on customer TW assignments to decide if new customers should be accepted and which TWs should be assigned. The authors discuss several data features and demonstrate that their data-driven approach outperforms a myopic policy. Interestingly, the computational experiments also show that using more historical data does not necessarily improve the results.

Another related area of problems are those with dynamically requesting customers and overlapping capture and fulfillment phases, e.g., in same-day delivery. Voccia et al. (2019) study a problem in which customers' requests appear over time, but, in contrast to Campbell and Savelsbergh (2005, 2006), the provider receives and fulfills requests within the same time horizon (i.e., same-day delivery). The objective is to maximize the number of served customers. The authors propose a Multiple-Scenario Approach (MSA).

Klein and Steinhardt (2023) study a same-day delivery problem in which the service provider offers delivery options, i.e., a combination of TWs and prices. Customers' behavior is uncertain, and they can reject the delivery options or accept one based on their utility. The probabilities associated with customer's choices are calculated using different utility models. As Voccia et al. (2019), the authors develop an MSA for their problem. A similar setting is considered in Abdollahi et al. (2023). The authors combine order forecasts and opportunity cost approximations to decide on the time slots and prices to offer to customers.

Most of the works dealing with dynamic customer requests and acceptance decisions assume that, if a customer is accepted, the service provider must ensure that the customer is served within the assigned TW. In other words, accepted requests cannot be canceled or moved. Checking whether it is possible to serve the new customer in a TW considering the already accepted customers is therefore crucial. The work of van der Hagen et al. (2022) focuses on this aspect in the context of attended home delivery. Instead of using a traditional heuristic algorithm, the authors propose a machine-learning approach to predict if serving a customer in a certain TW is feasible. Although their approach does not guarantee feasibility, computational results demonstrate that, on large-scale problems, the machine-learning approach performs better than insertion algorithms.

3. Problem Definition

In this section, we define the problem. We first provide an illustrative narrative of the problem and an example. We then model the problem as a sequential decision process.

3.1. Narrative

We take the perspective of a service provider serving customers at their homes. The provider operates in a service region with a fleet of service vehicles. The vehicles start their tours from a depot and are constrained by daily working times. All service and travel times are known and deterministic.

The set of potential customers is known but large, e.g., it comprises all possible addresses in the city. Requests for service occur over time in a capture phase and the service takes place in a fulfillment phase on another day in the future. For example, the customer requests are captured today, and fulfillment takes place the next day. Whenever a customer requests service, the provider assigns a TW of a set of potential TWs to the customer. The provider may also decide not to serve the customer, e.g., because of infeasibility. This customer then leaves the system (in practice, the customer is likely assigned a TW on another day for service, but this is outside of the scope of our work). A decision is feasible if, in the fulfillment phase, a routing can be found that serves all previous requests and the new request if a TW is assigned so that the vehicles can return to the depot in time. The objective is to find a policy that maximizes the number of successfully assigned TWs and therefore the served customers in the fulfillment phase.

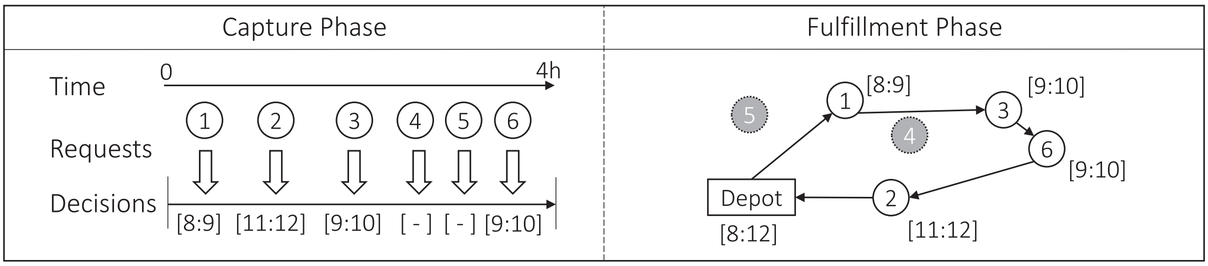

Figure 1 Example of the process.

3.2. Example

In the following, we show two examples. The first example illustrates the full decision process. The second example focuses on a specific state of the process in detail.

The process for one day is depicted in Figure 1. For this small example, we assume that in the fulfillment phase, the service provider has access to one vehicle with four hours of working time from 8 a.m. to 12 a.m. Furthermore, the provider can assign one-hour TWs ($[8:9]$, $[9:10]$, $[10:11]$, $[11:12]$) or no TW, indicated by $[-]$. On the left side of Figure 1, the process of the capture phase is depicted. We assume that the capture phase in which customers can request is four hours long. Here, six customer requests are revealed subsequently. For every customer, at the time of the request, a decision has to be made about which TW to assign (if any). In the example, first, TW $[8:9]$ is assigned to the first customer. Then, TW $[11:12]$ is assigned to the second customer and, later, TW $[9:10]$ to the third customer. Customers 4 and 5 are not assigned a TW ($[-]$), e.g., due to infeasibility. Finally, the last customer is assigned TW $[9:10]$, and the process terminates. On the right side of Figure 1, the routing of the fulfillment phase is shown. For the purpose of presentation, we omit the depiction of travel times in this example. On the day of fulfillment, the vehicle serves the first customer first, then customers 3, 6, and 2, and then returns to the depot within the four-hour working time. For illustration purposes, the figure also shows customers 4 and 5, even though they are not served that day, but another day - this is outside of the scope of the problem. We observe that customer 4 lies conveniently between customers 1 and 3. However, due to the assigned TWs for customers 1 and 3, an insertion at this position is infeasible. At the time of request 4, TW $[10:11]$ might have been feasible for this customer though. Since the assignment of this TW might have likely obstructed

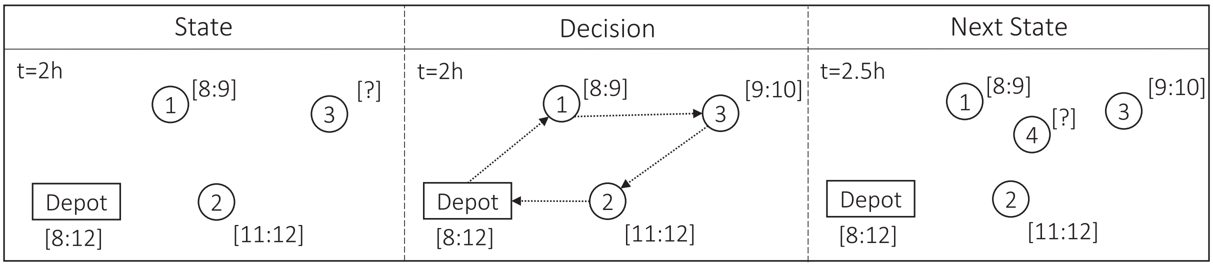

Figure 2 **Example of a state, decision, and transition to the next state.**

future services (e.g. at that time customer 6 is unknown), the provider decided against assigning this TW to customer 4 in this example. We also observe that assigning a TW to request 5 is infeasible due to the TWs assigned to customers 1 and 2.

We will now take a more detailed look at an individual decision state of the example to prepare the sequential decision process. We focus on the time, customer 3 requests service. The state is depicted on the left side of Figure 2. Customer 3 requests right in the middle of the capture phase, at $t = 2$h. At that time, two customers were already assigned TWs. The service provider now makes a decision about the TW to assign to customer 3. The decision depicted in this example is to assign TW $[9:10]$. No decisions about routing are made at that point. However, a TW-decision is feasible only if a feasible routing for the customers and their assigned TWs exists. A potential routing (D-1-3-2-D) is depicted by the dashed lines in the example. The reward of the decision is one because one additional customer could be added to the day. After a decision is made, the process continues. Here, at time $t = 2.5$h, the next customer requests, customer 4.

3.3. Sequential Decision Process

The sequence of states, decisions, revelation of new information, and transition to a new state can be modeled as a sequential decision process (Powell 2022a). In the following, we first present the global notation of the problem. Then, we define the components of the sequential decision process: decision points, state variables, decision variables and reward function, exogenous information, transition function, and objective function.

3.3.1. Global notation. We consider a fleet of m vehicles operating in a service area represented by the large set of locations $\mathcal{N}$. Vehicles start and end their tours at a depot denoted with $0 \in \mathcal{N}$

during the fulfillment phase horizon $[0, H_f]$. Travel time between locations $i, j \in \mathcal{N}$ is defined by the function t_{ij} and includes the service time at a customer i. Customers request a service over the capture phase horizon $[0, H_c]$. The finite set of TWs to offer for the next day (i.e., fulfillment phase horizon) is $\mathcal{T} = \{1, 2, \ldots, \tau, \tau+1, \ldots |\mathcal{T}|\}$. We indicate with $[e_\tau, \ell_\tau]$ the earliest and latest time at which the service can start if assigned to TW $\tau \in \mathcal{T}$. In our method and experiments, we assume that the entire service horizon is covered by the TWs, $\bigcup_{\tau \in \mathcal{T}} [e_\tau, \ell_\tau] = [0, H_f]$. In Section A.1 of the Appendix, we discuss the extension of our method if this is not the case.

3.3.2. Decision points. A decision point occurs when a new customer requests service. The decision points are denoted as $k \in \{0, 1, \ldots, K\}$, when the k^{th} customer request occurs. The number of decision points per day K is a random variable.

3.3.3. State variables. A state variable captures all information available when making a decision. For our problem, a state S_k at decision point k contains the following information:

- t_k: the time of the new request.
- i_k^{new}: the customer information of the k^{th} request.
- N_k: the set of customers that placed the order before decision point k and whose orders have been accepted, $N_k \subseteq \{1, 2, \ldots, k-1\}$. The number of accepted customers is $n_k = |N_k|$.
- $\mathcal{W}_k$: vector of n_k components storing the TWs assigned to the accepted requests. Component j, denoted by $\mathcal{W}_{jk}$, is set to the TW $\tau \in \mathcal{T}$ assigned to the accepted request $j \in N_k$.

In summary, the state S_k at decision point k is $S_k = (t_k, N_k, \mathcal{W}_k, i_k^{\text{new}})$ where the initial state S_0 is $S_0 = (0, \emptyset, \emptyset, 0)$.

3.3.4. Decision variables and reward function. At decision point k, the decision $x_k \in X(S_k) \subset \mathcal{T} \cup \{0\}$ is whether or not to accept the request of the new customer i_k^{new} and, in case of acceptance, what TW to assign. If the request is declined (i.e., $x_k = 0$), it is disregarded for the rest of the process. If the request is accepted (i.e., $x_k \in \mathcal{T}$), then one of the possible TWs of the new customer must be selected. We have to ensure that, if the request is accepted, there exists a feasible solution of the VRP with TWs involving the new customer and the previously accepted customers of the set N_k.

The reward of each decision is defined via function $R(S_k, x_k)$. The reward is $R(S_k, x_k) = 1$ if a TW is assigned to the customer (i.e., $x_k \in \mathscr{T}$) and zero otherwise (i.e., $R(S_k, 0) = 0$).

The combination of state and decision leads to a post-decision state $S_k^x = (t_k, N_k^x, \mathscr{W}_k^x)$. If the customer i_k^{new} was assigned a TW $\tau \in \mathscr{T}$, then $N_k^x = N_k \cup \{i_k^{\text{new}}\}$ and $\mathscr{W}_k^x = \mathscr{W}_k \oplus \{\tau\}$. Otherwise, $N_k^x = N_k$ and $\mathscr{W}_k^x = \mathscr{W}_k$.

3.3.5. Exogenous information. After decision x_k, exogenous information ω_{k+1} is revealed. The information consists of either a new customer request i_{k+1}^{new} at time t_{k+1} or the termination of the process (when no more customer requests paper).

3.3.6. Transition function. Based on post-decision state S_k^x and the exogenous information $\omega_{k+1} = \{i_{k+1}^{\text{new}}\}$, the transition function leads to a new state $S_{k+1} = (t_{k+1}, N_k^x, \mathscr{W}_k^x, i_{k+1}^{\text{new}})$

Alternatively, when the exogenous information does not provide a new request, the transition function leads to the termination state.

3.3.7. Objective function. A solution to our problem is a decision policy $\pi \in \Pi$, over all possible decision policies Π, defining a decision rule X^π that maps each state S_k to decision $x_k = X^\pi(S_k)$. As each decision x_k in state S_k contributes a revenue $R(S_k, x_k)$, the objective function of the problem is to find a policy π^* that maximizes the revenue over all decision points, i.e.,

$$\pi^* = \underset{\pi \in \Pi}{\arg\max} \Big[\sum_{k=0}^{K} R(S_k, X^\pi(S_k)) \mid S_0 \Big].$$

4. Solution Methodology

In this section, we present our stochastic lookahead. We will first give a general overview and motivation and then present the individual steps in detail.

4.1. Motivation and Overview

An effective policy needs to balance the immediate reward (assigning a TW to the customer now), and the expected future reward (being able to assign TWs to many customers in the future). This balance is captured in the famous Bellman Equation

$$\max_{x \in X(S_k)} R(S_k, x) + \mathbb{E}\Big[\sum_{l=k+1}^{K} R(S_l, X^{\pi^*}(S_l)) \mid S_k^x \Big].$$

The second term is also known as the *value* $V(S_k^x)$ of a post-decision state. Knowing the value function allows us to derive the optimal decision via the Bellman Equation. For problems of realistic dimensions, this is not possible due to the curses of dimensionality in state, decision, and information space. For our problem, the decision space is comparably small as only a TW from a reasonably sized set of potential TWs has to be communicated. However, TW-decisions are intertwined with detailed routing to confirm feasibility in current and future states. Furthermore, the state and information spaces are very large with customers potentially requesting all over the service area. This makes estimating the value function challenging. This challenge can be seen even in the small example in Figure 2. The decision about which TW to assign to request 3 (if any) is not obvious because the future requests and how the possible routing would look like are unknown. Many detailed factors come into play, including the location of the other requests and their TWs as well as the likelihood of future requests in the different parts of the service area.

In essence, a method that can capture this high level of detail in the approximation of the value function is required. There are essentially two different types of methods for approximating value functions (Soeffker et al. 2022): (1) offline methods such as value function approximation that learn the values for highly aggregated post-decisions states and (2) online methods such as stochastic lookaheads that perform their evaluation for each individual post-decision state observed during the decision process. As Ulmer et al. (2019) show, lookahead methods are better suited for problems that require very detailed considerations of the post-decision states, as for our problem. Thus, we propose a Stochastic Lookahead (SL).

In a state, our SL-method generates a set of scenarios. Each scenario extends the state by a set of future requests. Then, for every TW-decision, our method checks feasibility first by solving a VRP with TWs for a feasible solution. If the decision is feasible, we take the corresponding post-decision state and, for each scenario, create a TOP with Time Windows and Mandatory Visits (TOP-TW-MV) for the existing customers. Each of these problems is still NP-hard. Literature on stochastic dynamic routing proposes solving the scenarios heuristically to obtain feasible solutions in the scenarios (see,

Algorithm 1: Procedure of the Stochastic Lookahead

```
1  for each decision epoch (i.e., revealed customer) do
2      check the feasibility of the TWs for the new customer;
3      if there exists at least a feasible TW for the new customer then
4          generate scenarios;
5          for each possible decision (TWs + rejection) for the new customer do
6              for each scenario do
7                  obtain value;
8          make decision;
9      else
10         reject customer request;
```

e.g., Bent and Van Hentenryck 2004, Hvattum et al. 2007, Schilde et al. 2014, Sarasola et al. 2016, Ulmer et al. 2019, Voccia et al. 2019, Ulmer and Thomas 2020, Yu et al. 2023). However, heuristics may not provide accurate approximations given the complex problem to solve in this paper. As we need to solve many of them to obtain a good value approximation, we solve the linear relaxation of the corresponding problem instead. For a better approximation, we solve the linear relaxation of a strong reformulation of our problem. Once all relaxations are solved, we then count the additional requests we could serve for each scenario and set the value of the post-decision state as the average over all scenarios. We consider travel time as a tie-breaker in the case of several equally valuable decisions.

The procedure is visualized in Algorithm 1. The algorithm starts with a feasibility test (Step 2) to check whether it is possible to serve the customer or not in any TW. If it is impossible to serve the customer, the request is immediately rejected (Step 10). If serving the customer is feasible, the algorithm estimates the reward stemming from accepting the customer (for each TW) and rejecting it by using predictions (scenarios) for future potential customers (Step 4 – 7). In the final phase of the algorithm, the *feasible* decision with the highest reward is selected (Step 8). We note that due to the structure of the algorithm, parallelization in TWs and scenarios is possible. With parallelization, the average runtime per state is below one minute for all but one instance setting (See Table A2 in the Appendix). We detail the steps of Algorithm 1 in the following section.

4.2. Detailed Description of Algorithm 1

In this section, we provide details of every step of our algorithm.

4.2.1. Check the feasibility of the TWs for the new customer. Each time a new customer requests a service, the algorithm must assess if serving the customer is feasible. A customer i_k^{new} can be served if and only if there exists a set of m routes such that (a) each customer $j \in N_k$, i.e., every customer accepted in the previous epochs, can be served within TW $\mathscr{W}_{jk}$, and (b) customer i_k^{new} can be served within at least one TW $\tau \in \mathscr{T}$. If such a set of paths does not exist, customer i_k^{new} must be rejected.

Verifying the existence of these routes is NP-complete as it requires assessing the feasibility of a VRP with TWs in which (a) the set of customers that must be served is defined as $N = N_k \bigcup \{i_k^{\text{new}}\}$, (b) each customer $j \in N_k$ must be served within its assigned TW $\mathscr{W}_{jk}$, and (c) customer i_k^{new} must be served within the horizon $\left[0, H_f\right]$.

This problem can be represented on a directed graph $G = (V, A)$, where V is the set of all vertices, i.e., $V = N \bigcup \{0\}$ and the arc set is defined as $A = \{(i, j) | i, j \in V : i \neq j\}$. The earliest and latest time at which a service assigned to the TW $\tau \in \mathscr{T}$ can start are represented with e_τ and ℓ_τ, respectively, while m denotes the number of available vehicles. By introducing variables $v_{ij} \in \{0, 1\}$, equal to 1 if arc $(i, j) \in A$ is traversed by a vehicle, and $a_i \in \mathbb{R}_+$, equal to the arrival time at customer $i \in N$, the set of the feasible solutions serving customer i_k^{new} is defined as follows

$$\sum_{(i,j)\in A} v_{ij} = \sum_{(j,i)\in A} v_{ji} = 1 \qquad \forall i \in N \tag{1a}$$

$$\sum_{(0,j)\in A} v_{0j} = \sum_{(i,0)\in A} v_{i0} \leq m \tag{1b}$$

$$a_i + t_{ij} \leq a_j + H_f(1 - v_{ij}) \qquad \forall (i, j) \in A : i, j \in N \tag{1c}$$

$$\max\left\{t_{0i}, e_{\mathscr{W}_{ik}}\right\} \leq a_i \leq \min\left\{H_f - t_{i0}, \ell_{\mathscr{W}_{ik}}\right\} \qquad \forall i \in N_k \tag{1d}$$

$$t_{0i_k^{\text{new}}} \leq a_i \leq H_f - t_{i_k^{\text{new}}0} \tag{1e}$$

$$v_{ij} \in \{0, 1\} \qquad \forall (i, j) \in A \tag{1f}$$

$$a_i \in \mathbb{R}_+ \qquad \forall i \in N \tag{1g}$$

Constraints (1a) guarantee that every customer $i \in N$ is visited exactly once. Constraints (1b) ensure that at most m vehicles are used. Constraints (1c) are subtour elimination constraints in the Miller-Tucker-Zemlin form. Constraints (1d) and (1e) guarantee that each already accepted customer $i \in N_k$ is served within its assigned TW and that the new customer i_k^{new} is served within the planning horizon, respectively. Constraints (1f)–(1g) define the range of the decision variables.

Our feasibility check consists of solving (1) as an optimization problem using a solver with a time limit to reduce the computational effort and speed up our algorithm. If a solution is obtained within the time limit, serving the customer is feasible and the algorithm continues. If not, the customer is rejected. Although this heuristic feasibility check may reject customers for which a feasible TW exists, it ensures that if a customer is declared feasible, it can be assigned to at least one TW.

4.2.2. Generate scenarios. If it is feasible to accept customer i_k^{new}, the algorithm has to assess the *expected* reward of each decision concerning the new customer. The first phase of this procedure consists of generating a set Ω_k of $|\Omega_k|$, equiprobable, scenarios to estimate the impact of the current decision on future requests. The set Ω_k is generated by sampling "rest-of-the-day"-scenarios. Each scenario $\tilde{\omega} \in \Omega_k$ extends the existing state S_k by a set of sampled customer locations N_O. The locations and number of customers are drawn from the known spatial and temporal request probability distribution. Based on preliminary tests, we set the number of scenarios of our method and all benchmark methods to $|\Omega_k| = 10$.

4.2.3. Obtain value. The objective function of our sequential decision process, defined in Section 3.3.7, suggests that to obtain the value of the reward of a decision x_k, we need to estimate the number of additional future customers we can serve given the post-decision state corresponding to x_k. This results in solving a TOP-TW-MV (Lin and Vincent 2017) considering the set of scenarios generated in Step 4 to cope with the uncertain future requests.

Inspired by the relaxations proposed in Powell (2022b), Brown et al. (2010), Brown and Smith (2014), Balseiro and Brown (2019), we neglect all the non-anticipativy constraints. In other words, we assume that for each scenario the future customers are revealed all at the same time and not sequentially. This relaxation allows us to decompose the overall problem and assess the reward of a decision while solving a deterministic TOP-TW-MV for each scenario.

Because of this simplification, our procedure produces upper-bound values for each scenario. Thus, it likely overestimates the future services, and, in cases where problem resources are less scarce, may even lead to situations where all sampled customers in all scenarios can be served regardless of the assigned TW of the current customer. To cope with this issue we propose to include the routing time as a tie-breaker in the objective function of the TOP-TW-MV: if two solutions serve the same number of customers, we select the solution with the lowest routing time. The intuition behind this idea is that a solution with a lower routing time has additional time to accommodate more future customers. This plays a crucial role in Step 8, where the routing tie-breaker impacts the final decision.

In the next part of this section, we discuss the details regarding the definition and the formulations of the TOP-TW-MV. The TOP-TW-MV is a generalization of the well-known TOP with TWs (Kantor and Rosenwein 1992, Gunawan et al. 2016, Vansteenwegen et al. 2011) in which the set of customer N is partitioned into customers that must be served (*mandatory customers*) and customers that are served only if it is convenient and feasible (*optional customers*). The set of the mandatory customers N_M includes all the customers that placed an order up to epoch k whose orders have been accepted (N_k) and request i_k^{new} if $x_k \in \mathcal{T}$ (i.e., x_k consists of accepting i_k^{new} and assigning it to a TW). Formally, $N_M = N_k \bigcup \{i_k^{\text{new}}\}$ if $x_k \in \mathcal{T}$, $N_M = N_k$ otherwise. The set of the optional customer N_O corresponds to the set of requests in scenario $\tilde{\omega} \in \Omega_k$.

Each mandatory customer $j \in N_k$ must be served within the TW $\mathcal{W}_{jk}$ offered when the customer appeared, while i_k^{new} (if accepted) must be served within the TW x_k. Each optional customer $j \in N_O$ can be served in any TW $\tau \in \mathcal{T}$. To formulate this problem, we introduce the following decision variables: $v_{ij} \in \{0,1\}$, equal to 1 if arc $(i,j) \in A$ is traversed by a vehicle; $a_i \in \mathbb{R}_+$, equal to the arrival time at customer $i \in N$; and $y_i \in \{0,1\}$, equal to 1 if optional customer $i \in N_O$ is served and 0 otherwise. The problem can be formulated as follows

$$\rho(S_k, x_k, \tilde{\omega}) = \max \sum_{i \in N_O} y_i - \beta \sum_{(i,j) \in A} t_{ij} v_{ij} \tag{2a}$$

$$\text{s.t.} \sum_{(i,j)\in A} v_{ij} = \sum_{(j,i)\in A} v_{ji} = y_i \qquad \forall i \in N_O \qquad (2b)$$

$$\sum_{(i,j)\in A} v_{ij} = \sum_{(j,i)\in A} v_{ji} = 1 \qquad \forall i \in N_M \qquad (2c)$$

$$\sum_{(0,j)\in A} v_{0j} = \sum_{(i,0)\in A} v_{i0} \le m \qquad (2d)$$

$$a_i + t_{ij} \le a_j + H_f(1 - v_{ij}) \qquad \forall (i,j) \in A : i,j \in N \qquad (2e)$$

$$\max\left\{t_{0i}, e_{\mathcal{W}_{ik}}\right\} \le a_i \le \min\left\{H_f - t_{i0}, \ell_{\mathcal{W}_{ik}}\right\} \qquad \forall i \in N_M \qquad (2f)$$

$$t_{0i} y_i \le a_i \le \left(H_f - t_{i0}\right) y_i \qquad \forall i \in N_O \qquad (2g)$$

$$v_{ij} \in \{0,1\} \qquad \forall (i,j) \in A \qquad (2h)$$

$$y_i \in \{0,1\} \qquad \forall i \in N_O \qquad (2i)$$

$$a_i \in \mathbb{R}_+ \qquad \forall i \in N \qquad (2j)$$

The objective function (2a) maximizes the number of served optional customers minus the routing time multiplied by a factor β, set to $\frac{1}{(H_f+1)m}$, to ensure the routing part acts as a tie-breaker between solutions with the same number of served customers. Constraints (2b) ensure flow conservation at each optional customer while (2c) make that each mandatory customer must be served exactly once. Constraints (2d) limit the number of vehicles while (2e) eliminate subtours. Constraints (2f)–(2g) ensure each customer is visited within its TW (the assigned TWs for accepted customers and the horizon for i_k^{new}). Finally, constraints (2h)–(2j) define the domain of the variables.

In principle, one could solve problem (2) to assess the reward of a decision x_k at states S_k in a scenario $\tilde{\omega}$. However, even in its deterministic form, this problem is NP-hard because it can be reduced to a TOP-TW. Moreover, Algorithm 1 has to solve this problem several times. This is not doable in our setting: the dynamic nature of our problem requires quick evaluations since the decision concerning a request must be communicated in a reasonable time. This is a common issue in several dynamic problems involving routing decisions and this is one of the reasons why many works in this field rely on heuristics (e.g., Campbell and Savelsbergh 2005).

An alternative approach is to work with a simplified version of the problem by for example considering one of its possible relaxations. In our case, a straightforward choice is to work with the linear relaxation of (2). The advantage is that to assess the reward we now need to solve several linear programs, rather than several NP-hard problems. The drawback is that if the relaxation is loose, i.e. the gap between the optimal objective value of the relaxation and of the original problem is large, the assessed reward may be inaccurate and lead to low rewarding decisions.

To achieve a trade-off between computational time and accuracy, we propose to work with the linear relaxation of an alternative formulation of (2). In particular, we consider the set packing/partitioning reformulations of (2) described in the following.

Let $\mathscr{R}$ be the set of all elementary routes in graph G whose total duration does not exceed H_f and such that each visited customer is visited within one of the corresponding feasible TWs. For each route $r \in \mathscr{R}$, let p_r be the number of customers served in the route minus the routing time times β. Finally, let s_{ir} be an integer coefficient indicating how many times customer $i \in N$ is served in route $r \in \mathscr{R}$. Using a binary variable $\xi_r \in \{0,1\}$ for each route $r \in \mathscr{R}$ to represent the decision of selecting a route (i.e., $\xi_r = 1$) or not (i.e., $\xi_r = 0$), the TOP-MTW can be reformulated as follows

$$\rho(S_k, x_k, \tilde{\omega}) = \max \sum_{r \in \mathscr{R}} p_r \xi_r \tag{3a}$$

$$\text{s.t.} \sum_{r \in \mathscr{R}} s_{ir} \xi_r = 1 \qquad \forall i \in N_M \tag{3b}$$

$$\sum_{r \in \mathscr{R}} s_{ir} \xi_r \leq 1 \qquad \forall i \in N_O \tag{3c}$$

$$\sum_{r \in \mathscr{R}} \xi_r \leq m \tag{3d}$$

$$\xi_r \in \{0,1\} \qquad \forall r \in \mathscr{R} \tag{3e}$$

The objective function (3a) aims at maximizing the number of served customers. Constraints (3b) ensure that each mandatory customer is served exactly once. Constraints (3c) guarantees that each optional customer is served at most once. Constraint (3d) enforces that no more than m routes are selected. Constraints (3e) define the domain of the decision variables.

The advantage of formulation (3) over formulation (2) is the strength of its linear relaxation, as demonstrated in other studies on similar problems (e.g., Orlis et al. 2020). In other words, the dual bound obtained by solving the linear relaxation of (3) is typically an accurate (over) estimation of

$\rho(S_k, x_k, \tilde{\omega})$. We, therefore, use the linear relaxation of (3) to assess the reward of a decision x_k. We denote by $\hat{\rho}(S_k, x_k, \tilde{\omega})$ the dual bound obtained by solving the linear relaxation of (3) in state S_k for decision x_k and scenario $\tilde{\omega}$. Notice that as the feasibility check performed in Step 2 does not specify which TW is feasible, the linear relaxation of (3) can be infeasible for some x_k. In this case, we set $\hat{\rho}(S_k, x_k, \tilde{\omega}) = -\infty$ for each $\tilde{\omega} \in \Omega_k$.

Formulation (3) has an exponential number of decision variables and, therefore, we must solve its linear relaxation via column generation (Lübbecke and Desrosiers 2005). The master problem is initialized with the single-customer routes plus one artificial column with a properly defined profit to deal with infeasibility. The pricing problem is solved using dynamic programming and generating ng-routes instead of elementary routes (Baldacci et al. 2011, 2012).

4.2.4. Make decision. The final step of our algorithm consists of selecting the feasible solution x_k with the highest reward. The procedure starts with calculating the total reward of each solution $x_k \in \mathcal{T} \bigcup \{0\}$ over all the scenarios, then the algorithm tests the feasibility of the solutions in non-decreasing order of total assessed reward until a feasible solution is found.

The total assessed reward of a decision $x_k \in \mathcal{T}$ is given by

$$\frac{1}{|\Omega_k|} \sum_{\tilde{\omega} \in \Omega_k} \hat{\rho}(S_k, x_k, \tilde{\omega}), \tag{4}$$

where $\hat{\rho}(S_k, x_k, \tilde{\omega})$ is calculated in the previous steps of the algorithm for each scenario $\tilde{\omega} \in \Omega_k$. The best solution x_k^* is defined as

$$x_k^* = \underset{x_k \in \widetilde{\mathcal{T}}}{\arg\max} \left\{ \frac{1}{|\Omega_k|} \sum_{\tilde{\omega} \in \Omega_k} \hat{\rho}(S_k, x_k, \tilde{\omega}) \right\} \tag{5}$$

where $\widetilde{\mathcal{T}}$ is initially set equal to $\mathcal{T}$. Notice that $\hat{\rho}(S_k, x_k, \tilde{\omega})$ includes the routing times multiplied by factor β. Therefore, the routing time will act as a tie-breaker in the selection of the best solution. In other words, if multiple TWs achieve the maximum value, the algorithm selects the one with the lowest routing time.

Before accepting x_k^*, its feasibility is checked by solving an instance of (1) in which the set of available TWs for customer i_k^{new} contains only x_k^*, i.e., $\mathcal{T} = \{x_k^*\}$. As in Step 2, if the feasibility can not be proved within a time limit, the decision is considered infeasible. If x_k^* is feasible, the decision is accepted and S_{k+1} is updated accordingly. Otherwise, $\widetilde{\mathcal{T}} = \widetilde{\mathcal{T}} \setminus \{x_k^*\}$ and the procedure is repeated until a feasible decision is obtained from (5). It is worth noticing that TWs with high rewards tend to be feasible: intuitively if the solution of the relaxed problem visits all the mandatory customers plus many more optional ones (high reward), the chance that it is feasible to visit all the mandatory customers in the integer problem is higher. This property usually ensures that, if serving the current customer is feasible, the first offered TW (i.e., the one with the highest score) passes the feasibility check and the procedure stops after one iteration.

5. Experimental Study

In this section, we present our computational study. We first describe the instances and benchmark policies. We then analyze the performance of our method and its impact on service availability.

5.1. Instances

We test our methodology on instances based on Ulmer and Thomas (2019). The depot is set as in Ulmer and Thomas (2019). The instances contain data from more than 30,000 locations in Iowa City, Iowa. We uniformly sample days of requests from the locations. We consider one and two vehicles working 8-hour shifts. We set the expected number of requests to 30 for the one-vehicle and 50 for the two-vehicle case. Service times are set to 10 minutes per customer resembling delivery or easier technical tasks. The number of TWs is eight, each TW has a size of one hour. TWs are not overlapping and therefore cover the entire eight hours of the fulfillment phase. Distances between locations are calculated using the Haversine measure (Shumaker and Sinnott 1984). Iowa City is rather small with a diameter of less than 10 km. For many of the considered problems, service vehicles have to travel longer distances with widely spread customer locations. Thus, we use the assumed speed to control the geographical spread of customers. To this end, we set the speed to 25 km per hour (*small* spread), 15 km per hour (*regular* spread), and 10 km per hour (*large* spread). For each instance, we generate 100 realizations different from the scenarios used in the policies.

5.2. Benchmark Policies

We present two types of benchmark policies. First, we compare our method to practical policies to analyze its general effectiveness. Second, we compare our benchmark to other relaxation approaches to investigate the advantage of our formulation.

We present a variety of problem-specific benchmarks from the literature:

1. *Insertion*: This procedure maintains a tentative route. New requests are inserted in the route, at the cheapest feasible position. The TW for the new request is then set based on the tentative arrival time in the updated route.
2. *Anticipatory Insertion* (AI): This benchmark is based on the insights in Ulmer and Thomas (2019). The routing and insertion follow the same procedure as Insertion. However, instead of assigning the TW based on the current arrival time, the arrival times are scaled with respect to the current tour duration and overall time available. This enables the assignment of more distributed TWs and adds some slack for new customers along the route.
3. *Geography* (G): This procedure is derived from Sungur et al. (2010). It assigns static TWs to different areas of the city. To this end, we first create $|\mathscr{T}|$ clusters of potential customer locations via k-means. We then determine a "master tour" by solving the Traveling Salesperson Problem (TSP) through the cluster centers and assign TWs in the TSP-sequence. Thus, customers from the first cluster in the TSP-sequence get the first TW, from the second cluster the second, etc. We perform the feasibility check by solving an instance of problem (1) with the previously accepted customers and the new one.
4. *Multiple-Scenario Approach* (MSA): This idea is derived from Campbell and Savelsbergh (2005). Instead of solving the Bellman Equation to select the highest-value solution, we search for the most popular solution among the scenarios. Each time a new request appears, the procedure first checks its feasibility by looking for a primal bound using an insertion algorithm (we need to insert all the accepted customers in their TW plus the current request). If the request is feasible, we sample scenarios. For each scenario, we run an insertion algorithm. The insertion

algorithm inserts customers at minimum “cost" (route duration). It inserts first the accepted customers (to ensure feasibility). Then it looks for the best customer to insert among the current request and the sampled customers (the criterion is minimum route duration). It continues until is not possible to add more customers. If the current request has been inserted in the route more than half of the times (i.e., more than half of the number of scenarios), the current request is accepted; it is rejected otherwise. The TW is selected by running again the insertion algorithm to insert the current request.

Our methodology approximates the Bellman Equation via a dual bound on a mixed-integer program. To analyze the strength of the relaxation (3), we present two additional benchmarks:

1. *Stochastic Lookahead with Standard Bound* (SL-SB): The approach follows the same procedure described in Algorithm 1. However, in Step (7) (obtain value), instead of using the dual bound of formulation (3), we use the dual bound of the formulation (2).
2. *Stochastic Lookahead with Heuristic Bound* (SL-HB): This policy follows the general procedure of SL, but it solves the scenarios with a heuristic, namely, the cheapest insertion heuristic. Using heuristics to solve scenarios is the prevalent approach for stochastic lookahead methods in the stochastic dynamic VRP literature.

Finally, we also present a *Perfect Information upper Bound* (PIB) solution. We consider a deterministic problem where all the requests are known before assigning the TWs. For this instance, we compute the linear relaxation of (3).

5.3. Policy Performance

Figure 3 summarizes the experimental results by depicting the average objective value of all policies over all instance settings - the individual results can be found in the Appendix. The x-axis shows the policy, and the y-axis the average objective value. The light bar represents the perfect information upper bound. The dark blue/grey bar represents our policy SL. The blue/grey bars represent problem-oriented benchmark policies, and the light blue/grey bars represent method-oriented benchmark policies.

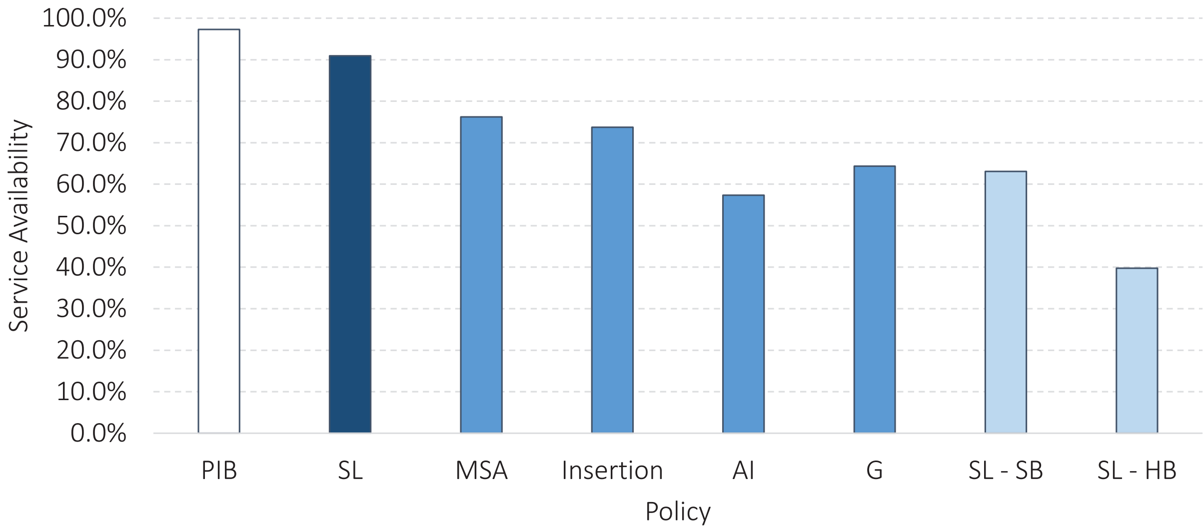

Figure 3 **Objective values for the upper bound (light), our policy SL (dark blue/grey), and all problem- (blue/grey) and method-oriented (light blue/grey) benchmark policies.**

We observe that our policy outperforms all problem- and method-oriented benchmarks and achieves results close to the perfect information upper bound. Thus, our policy performs quite effectively compared to the external state-of-the-art benchmark policies. Interestingly, the rather simple Insertion-policy performs nearly as well as the MSA and significantly better than the Anticipatory Insertion-policy. This might be explained by the fact that AI assumes an equal probability of customer occurrences along the route which is in contrast to the very heterogeneous customer distribution in Iowa City.

We also observe that statically assigning TWs to regions in the city done by the Geography-policy (G) performs slightly worse than Insertion. Interestingly, while the two policies do not differ much in solution quality, their decision-making is very different as we show later in this section. Equally noteworthy is the poor performance of the alternative SL-methods. The standard relaxation SL-SB is not able to outperform the simple Insertion-strategy. The most prominent approach from the literature, i.e., the heuristic bound method SL-HB performs even worse.

The relative performance of all policies is the same over all instance settings as shown in Table 1. Our policy achieves the best results for all individual instance settings. As expected though, the general solution quality decreases with the increasing spread of customers. We also note that the

Table 1 Performance of all policies on all the benchmarks

Num. Veh./Req.	Spread	PIB	SL	MSA	Insertion	AI	G	SL - SB	SL - HB
1/30	large	88.3%	78.9%	51.3%	45.4%	42.9%	40.4%	46.0%	28.9%
	regular	96.4%	90.5%	64.5%	61.6%	54.7%	60.8%	62.1%	32.7%
	small	99.6%	96.0%	85.4%	84.5%	77.5%	83.0%	82.7%	37.2%
2/50	large	99.4%	84.6%	69.7%	65.6%	43.3%	47.1%	60.6%	43.1%
	regular	100.0%	95.8%	87.6%	86.5%	51.4%	65.0%	57.7%	43.0%
	small	100.0%	99.8%	98.7%	98.7%	74.4%	89.8%	69.5%	53.7%

routing tie-breaker is particularly valuable for cases where resources are less scarce, i.e., cases with two vehicles or small spread. The corresponding results can be found in Table A1.

5.4. Service Availability

Finally, we analyze the impact of our policy on the customer experience. Previous research has shown that when deciding about service with respect to the Bellman Equation, this may often come at the expense of "inconvenient" customer. Such customers may be feasible but may consume lots of time to serve. Often, they can be found in the outskirts of the city (Soeffker et al. 2017, Chen et al. 2023). Thus, even when they request service early, they may not be served. Such active discrimination against some customers leads to several drawbacks. Besides public outcry, it becomes particularly disadvantageous in cases where rejected customers have to be served on a later day (Ulmer et al. 2018). To analyze potential discrimination, we analyze the spatial and temporal service availability for the setting with one vehicle, a regular spread of customers, and 30 expected requests. For illustration purposes, we compare our policy to the practical benchmarks Insertion and Geography.

We first analyze the temporal service availability over the requests. This is shown in Figure 4. The x-axis depicts the request number, and the y-axis the average service availability for the request number. Because the expected number of requests is 30, the values for requests 30-50 are less stable because they are observed less frequently. The solid line represents our SL-policy, the dashed line the Insertion-policy, and the dotted line the Geography-policy. For all of them, we observe a decline

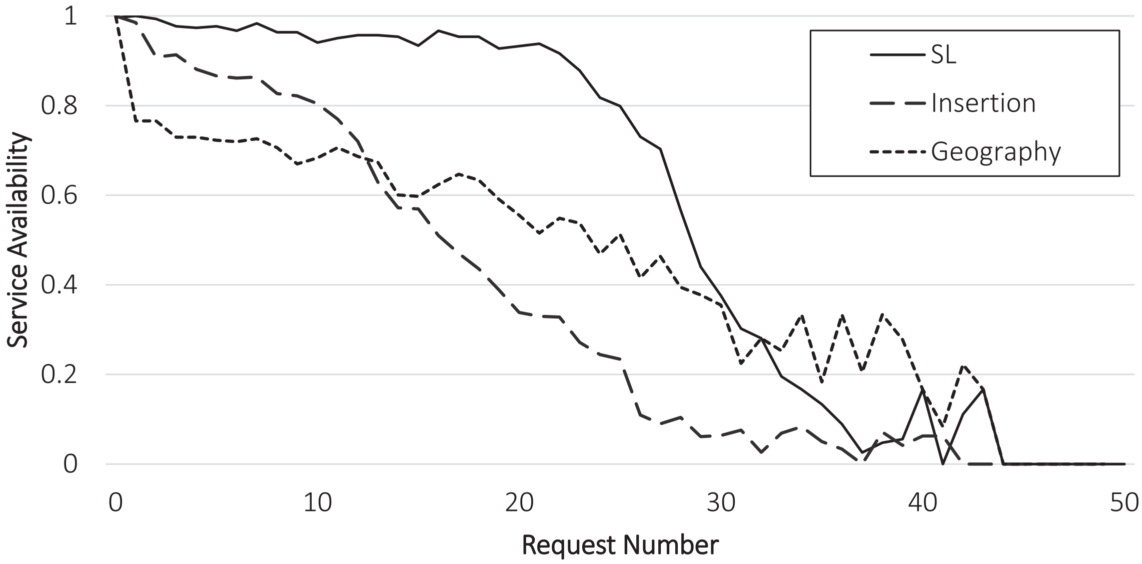

Figure 4 **Service availability over the request number for SL, Insertion, and Geography.**

in service availability over the request numbers. That can be expected. Customers requesting later are less likely to be accepted, simply, because the vehicle capacities are consumed.

However, we observe significant differences in the shape of the curves. The Geography-policy shows a smaller service rate in the beginning followed by a less severe decline. This policy assigns each part of the service area exactly one TW based on a master tour through the center of the parts. It then only checks, if the corresponding TW for a request is feasible. If not, the "inconvenient" request is declined even if an alternative TW would be feasible. This selective procedure and the "insistence" of the master tour saves some resources such that service might be available even for some late requests. In contrast, for the Insertion-policy, initially, all customers can be offered TWs because of the routing flexibility. Afterward, due to the missing anticipation, poor TW-decisions lead to infeasibility early in the process. This is indicated by the steep decline after the 10th request. Our SL-policy combines both advantages, routing flexibility and anticipation. Therefore, it achieves a very high service availability for many requests. This result not only confirms the superiority of the SL-policy but also indicates that, in contrast to the Geography-policy, our policy does not reject "inconvenient" requests even though they might be feasible.

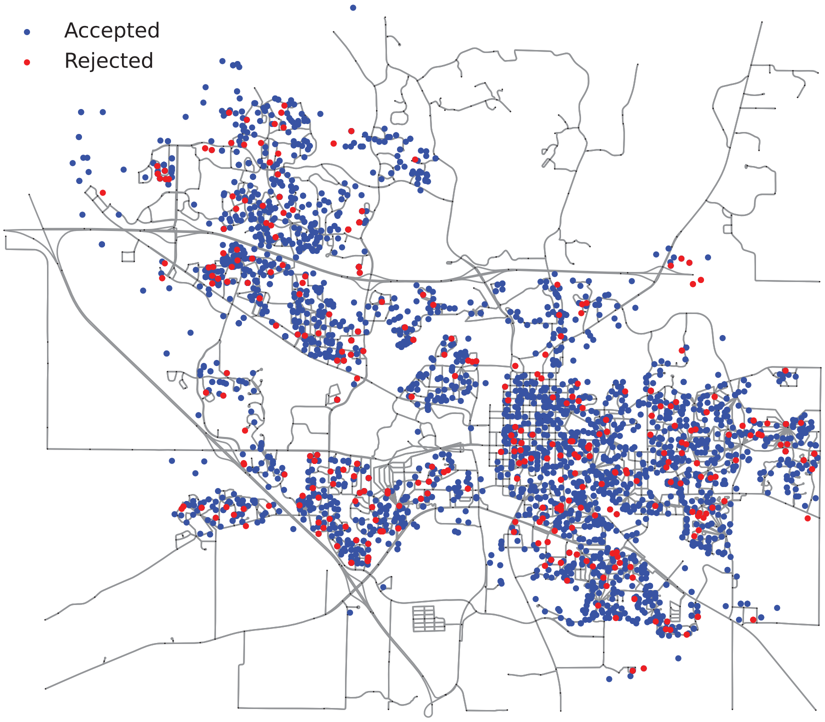

Figure 5 Locations of accepted and rejected customers for SL.

This is confirmed by analyzing the spatial distribution of accepted and rejected customers in Figure 5. The red (light gray) dots indicate rejections, and the blue (dark grey) dots indicate acceptances. No clear structure can be observed, the acceptances and rejections are rather randomly distributed in the service area. This again confirms that our SL-policy achieves a high service availability even for "inconvenient" customers.

6. Conclusion

In this work, we have shown how tight relaxations of scenarios in combination with the Bellman Equation enable fast and detailed anticipation and overall high customer service quality. There are several avenues for future research with respect to the problem and method.

For the problem at hand, we were able to assign narrow TWs without reducing the service level substantially. However, Ulmer et al. (2023) showed that for a few customers, e.g., the first customers of the day, assigning a single, narrow TW can reduce flexibility significantly. The problem may therefore be extended to either assign differently sized TWs or a set of TWs in which the service can take place. The proposed methodology can be applied directly to the first, and with slight adaption, to the second extension of the problem. In our problem and experiments, we focused on one source of uncertainty, stochastic requests. As Ulmer and Thomas (2019) showed, this source is substantially more disruptive than others such as stochastic travel and service times. However, during operations, severe traffic may lead to missed TWs or even to dynamic rerouting - in both cases, TWs may be updated during the process. Future research may therefore extend our work to cases of uncertainty during the fulfillment phase. For stochastic travel times, the work by Ehmke et al. (2015) may be embedded to evaluate the risk of violating an assigned TW. For dynamic rerouting, our work may be paired with the idea of Dalmeijer et al. (2019) of dynamic TW-updates. Finally, the problem horizon may be extended to multiple days as suggested in Avraham and Raviv (2021). Decisions would not include the "no TW"-decision and would be made not only about the TW in a day but about the day itself. This may also change the objective from expected services per day to average waiting time for service per customer.

There are further several potential extensions of our methodology. While our method allows for real-time decision making within minutes, future research may investigate further means to reduce calculation time. For example, the time-consuming feasibility checks for each TW might be supported by means of supervised learning as suggested in van der Hagen et al. (2022). Further, our methodology assumes no TW-decisions for the sampled customers in the scenarios which might lead to overly optimistic evaluations. Random TWs would likely lead to unrealistically low values. Ideally, one or more TWs would be assigned that come close to realistic TW-decisions. While we do not discuss it explicitly in this paper, we tested several approaches to assign TWs to the sampled customers, e.g., policies suggested in Köhler et al. (2020) where we assigned the TWs of the closest real customers. However, none of the approaches led to visible improvements compared to our proposed policy. Here, similar to Ulmer et al. (2019), a policy may be trained via reinforcement learning and then used in the creation of the scenarios.

Finally, while we designed our method to solve the TW assignment problem at hand, the general challenge of anticipating complex future routing decisions in current real-time decision-making is omnipresent in stochastic dynamic routing (Soeffker et al. 2022, Hildebrandt et al. 2023). Future research may therefore transfer the general concept of our method to other dynamic routing problems, e.g., stochastic dynamic pickup and delivery problems such as same-day or meal delivery.

Acknowledgments

Marlin Ulmer's work is funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) Emmy Noether Programme, project 444657906. We gratefully acknowledge their support.

References

M. Abdollahi, X. Yang, M. I. Nasri, and M. Fairbank. Demand management in time-slotted last-mile delivery via dynamic routing with forecast orders. *European Journal of Operational Research*, 309(2):704–718, 2023.

E. Avraham and T. Raviv. The steady-state mobile personnel booking problem. *Transportation Research Part B: Methodological*, 154:266–288, 2021.

R. Baldacci, A. Mingozzi, and R. Roberti. New route relaxation and pricing strategies for the vehicle routing problem. *Operations Research*, 59(5):1269–1283, 2011.

R. Baldacci, A. Mingozzi, and R. Roberti. New state-space relaxations for solving the traveling salesman problem with time windows. *INFORMS Journal on Computing*, 24(3):356–371, 2012.

S. R. Balseiro and D. B. Brown. Approximations to stochastic dynamic programs via information relaxation duality. *Operations Research*, 67(2):577–597, 2019.

R. W. Bent and P. Van Hentenryck. Scenario-based planning for partially dynamic vehicle routing with stochastic customers. *Operations Research*, 52(6):977–987, 2004.

D. B. Brown and J. E. Smith. Information relaxations, duality, and convex stochastic dynamic programs. *Operations Research*, 62(6):1394–1415, 2014.

D. B. Brown, J. E. Smith, and P. Sun. Information relaxations and duality in stochastic dynamic programs. *Operations Research*, 58(4-part-1):785–801, 2010.

A. M. Campbell and M. Savelsbergh. Incentive schemes for attended home delivery services. *Transportation Science*, 40(3):327–341, 2006.

A. M. Campbell and M. W. P. Savelsbergh. Decision support for consumer direct grocery initiatives. *Transportation Science*, 39(3):313–327, 2005.

S. Celik, L. Martin, A. H. Schrotenboer, and T. van Woensel. Exact two-step benders decomposition for two-stage stochastic mixed-integer programs. *arXiv preprint arXiv:2306.02849*, 2023.

X. Chen, T. Wang, B. W. Thomas, and M. W. Ulmer. Same-day delivery with fair customer service. *European Journal of Operational Research*, 308(2):738–751, 2023.

K. Dalmeijer and G. Desaulniers. Addressing orientation symmetry in the time window assignment vehicle routing problem. *INFORMS Journal on Computing*, 33(2):495–510, 2021.

K. Dalmeijer and R. Spliet. A branch-and-cut algorithm for the time window assignment vehicle routing problem. *Computers & Operations Research*, 89:140–152, 2018.

K. Dalmeijer, R. Spliet, and A. Wagelmans. Dynamic time window adjustment. Technical Report EI2019-22, Econometric Institute, Erasmus School of Economics, Erasmus University Rotterdam, May 2019. URL `http://hdl.handle.net/1765/116533`.

J. F. Ehmke and A. M. Campbell. Customer acceptance mechanisms for home deliveries in metropolitan areas. *European Journal of Operational Research*, 233(1):193–207, 2014.

J. F. Ehmke, A. M. Campbell, and T. L. Urban. Ensuring service levels in routing problems with time windows and stochastic travel times. *European Journal of Operational Research*, 240(2):539–550, 2015.

B. Ellis. Waiting for the cable guy is costing us $38 billion, 2011. URL `http://money.cnn.com/2011/11/03/pf/cost_of_waiting/`. [Online; accessed 23-October-2023].

D. Fleckenstein, R. Klein, and C. Steinhardt. Recent advances in integrating demand management and vehicle routing: A methodological review. *European Journal of Operational Research*, 306(2):499–518, 2023.

J. Flinterman, L. Martin, J. Kinable, A. Schrotenboer, A. Schuffelers, and P. Hulleman. Assigning delivery time windows before all customers are known. Master's thesis, Department of Industrial Engineering & Innovation Sciences, 2022.

A. Gunawan, H. C. Lau, and P. Vansteenwegen. Orienteering problem: A survey of recent variants, solution approaches and applications. *European Journal of Operational Research*, 255(2):315–332, 2016.

F. D. Hildebrandt, B. W. Thomas, and M. W. Ulmer. Opportunities for reinforcement learning in stochastic dynamic vehicle routing. *Computers & Operations Research*, 150:106071, 2023.

L. M. Hvattum, A. Løkketangen, and G. Laporte. A branch-and-regret heuristic for stochastic and dynamic vehicle routing problems. *Networks*, 49(4):330–340, 2007.

O. Jabali, R. Leus, T. van Woensel, and T. de Kok. Self-imposed time windows in vehicle routing problems. *OR Spectrum*, 37:331–352, 2015.

M. G. Kantor and M. B. Rosenwein. The orienteering problem with time windows. *Journal of the Operational Research Society*, 43(6):629–635, 1992.

V. Klein and C. Steinhardt. Dynamic demand management and online tour planning for same-day delivery. *European Journal of Operational Research*, 307(2):860–886, 2023.

C. Köhler, J. F. Ehmke, and A. M. Campbell. Flexible time window management for attended home deliveries. *Omega*, 91:102023, 2020.

C. Köhler, A. M. Campbell, and J. F. Ehmke. Data-driven customer acceptance for attended home delivery. *OR Spectrum*, pages 1–36, 2023a.

C. Köhler, J. F. Ehmke, A. M. Campbell, and C. Cleophas. Evaluating pricing strategies for premium delivery time windows. *EURO Journal on Transportation and Logistics*, 12:100108, 2023b.

S.-W. Lin and F. Y. Vincent. Solving the team orienteering problem with time windows and mandatory visits by multi-start simulated annealing. *Computers & Industrial Engineering*, 114:195–205, 2017.

M. E. Lübbecke and J. Desrosiers. Selected topics in column generation. *Operations Research*, 53(6):1007–1023, 2005.

F. Neves-Moreira, D. P. da Silva, L. Guimarães, P. Amorim, and B. Almada-Lobo. The time window assignment vehicle routing problem with product dependent deliveries. *Transportation Research Part E: Logistics and Transportation Review*, 116:163–183, 2018.

C. Orlis, N. Bianchessi, R. Roberti, and W. Dullaert. The team orienteering problem with overlaps: An application in cash logistics. *Transportation Science*, 54(2):470–487, 2020.

W. Powell. *Reinforcement Learning and Stochastic Optimization*. John Wiley & Sons, Ltd, 2022a. ISBN 9781119815068. doi: https://doi.org/10.1002/9781119815068.

W. B. Powell. Designing lookahead policies for sequential decision problems in transportation and logistics. *IEEE Open Journal of Intelligent Transportation Systems*, 3:313–327, 2022b.

B. Sarasola, K. F. Doerner, V. Schmid, and E. Alba. Variable neighborhood search for the stochastic and dynamic vehicle routing problem. *Annals of Operations Research*, 236:425–461, 2016.

M. Schilde, K. F. Doerner, and R. F. Hartl. Integrating stochastic time-dependent travel speed in solution methods for the dynamic dial-a-ride problem. *European Journal of Operational Research*, 238(1):18–30, 2014.

B. P. Shumaker and R. W. Sinnott. Astronomical computing: 1. Computing under the open sky. 2. Virtues of the haversine. *Sky and Telescope*, 68(2):158–159, 1984.

N. Soeffker, M. W. Ulmer, and D. C. Mattfeld. On fairness aspects of customer acceptance mechanisms in dynamic vehicle routing. *Proceedings of Logistikmanagement*, 2017:17–24, 2017.

N. Soeffker, M. W. Ulmer, and D. C. Mattfeld. Stochastic dynamic vehicle routing in the light of prescriptive analytics: A review. *European Journal of Operational Research*, 298(3):801–820, 2022.

R. Spliet and G. Desaulniers. The discrete time window assignment vehicle routing problem. *European Journal of Operational Research*, 244(2):379–391, 2015.

R. Spliet and A. F. Gabor. The time window assignment vehicle routing problem. *Transportation Science*, 49 (4):721–731, 2015.

R. Spliet, S. Dabia, and T. Van Woensel. The time window assignment vehicle routing problem with time-dependent travel times. *Transportation Science*, 52(2):261–276, 2018.

A. Subramanyam, A. Wang, and C. E. Gounaris. A scenario decomposition algorithm for strategic time window assignment vehicle routing problems. *Transportation Research Part B: Methodological*, 117:296–317, 2018.

I. Sungur, Y. Ren, F. Ordóñez, M. Dessouky, and H. Zhong. A model and algorithm for the courier delivery problem with uncertainty. *Transportation Science*, 44(2):193–205, 2010.

M. W. Ulmer and B. W. Thomas. Enough waiting for the cable guy — Estimating arrival times for service vehicle routing. *Transportation Science*, 53(3):897–916, 2019.

M. W. Ulmer and B. W. Thomas. Meso-parametric value function approximation for dynamic customer acceptances in delivery routing. *European Journal of Operational Research*, 285(1):183–195, 2020.

M. W. Ulmer, N. Soeffker, and D. C. Mattfeld. Value function approximation for dynamic multi-period vehicle routing. *European Journal of Operational Research*, 269(3):883–899, 2018.

M. W. Ulmer, J. C. Goodson, D. C. Mattfeld, and M. Hennig. Offline–online approximate dynamic programming for dynamic vehicle routing with stochastic requests. *Transportation Science*, 53(1):185–202, 2019.

M. W. Ulmer, J. C. Goodson, and B. W. Thomas. Optimal service time windows. *Working Paper Series*, 2023.

L. van der Hagen, N. Agatz, R. Spliet, T. R. Visser, and A. L. Kok. Machine learning-based feasibility checks for dynamic time slot management. *Transportation Science*, Articles in Advance, 2022.

P. Vansteenwegen, W. Souffriau, and D. Van Oudheusden. The orienteering problem: A survey. *European Journal of Operational Research*, 209(1):1–10, 2011.

S. A. Voccia, A. M. Campbell, and B. W. Thomas. The same-day delivery problem for online purchases. *Transportation Science*, 53(1):167–184, 2019.

S. Voigt, M. Frank, P. Fontaine, and H. Kuhn. The vehicle routing problem with availability profiles. *Transportation Science*, 57(2):531–551, 2023.

K. Waßmuth, C. Köhler, N. Agatz, and M. Fleischmann. Demand management for attended home delivery – A literature review. *European Journal of Operational Research*, 311:801–815, 2023.

X. Yu, S. Shen, B. Badri-Koohi, and H. Seada. Time window optimization for attended home service delivery under multiple sources of uncertainties. *Computers & Operations Research*, 150:106045, 2023.

C. Zara. Comcast and Dish network roll out technician tracking apps in race to alleviate customer service frustrations, September 23 2015. URL `http://www.ibtimes.com/comcast-dish-network-roll-out-technician-tracking-apps-race-alleviate-customer-2110959`. [Online; accessed 23-October-2023].

Appendix A.1: Formulations for Problems with Multiple TWs

If the union of the TWs does not cover the entire horizon $[0, H_f]$, formulations (1) and (2) must be extended to accommodate multiple TWs for the new customer and optional customers, respectively.

Formulation (1) can be adapted by introducing variables z_τ, equal to 1 if TW $\tau \in \mathcal{T}$ is assigned to customer i_k^{new}. The problem becomes a VRP with Multiple TWs. Using the same parameters, notation, and variables of (1), the problem can be formulated as follows

$$\sum_{(i,j)\in A} v_{ij} = \sum_{(j,i)\in A} v_{ji} = 1 \qquad \forall i \in N \tag{A1a}$$

$$\sum_{(0,j)\in A} v_{0j} = \sum_{(i,0)\in A} v_{i0} \leq m \tag{A1b}$$

$$a_i + t_{ij} \leq a_j + H_f(1 - v_{ij}) \qquad \forall (i,j) \in A : i,j \in N \tag{A1c}$$

$$\max\{t_{0i}, e_{\mathcal{W}_{ik}}\} \leq a_i \leq \min\{\ell_{\mathcal{W}_{ik}}, H_f - t_{i0}\} \qquad \forall i \in N_k \tag{A1d}$$

$$\sum_{\tau\in\mathcal{T}} \max\left\{t_{0i_k^{\text{new}}}, e_\tau\right\} z_\tau \leq a_{i_k^{\text{new}}} \leq \sum_{\tau\in\mathcal{T}} \min\left\{\ell_\tau, H_f - t_{i_k^{\text{new}}0}\right\} z_\tau \tag{A1e}$$

$$\sum_{\tau\in\mathcal{T}} z_\tau = 1 \tag{A1f}$$

$$v_{ij} \in \{0,1\} \qquad \forall (i,j) \in A \tag{A1g}$$

$$a_i \in \mathbb{R}_+ \qquad \forall i \in N \tag{A1h}$$

$$z_\tau \in \{0,1\} \qquad \forall \tau \in \mathcal{T} \tag{A1i}$$

Constraints (A1a) - (A1d) correspond to constraints (1a)–(1d), i.e., they guarantee that every customer is visited once, bound the number of vehicles, avoid subtours, and impose TWs on the already accepted customers. Constraint (A1e) links the TW assigned to i_k^{new} and the corresponding arrival time of the vehicle. Constraint (A1f) ensures that exactly one TW is assigned to the new customer. Constraints (A1g)-(A1i) define the domain of the decision variables.

To adapt formulation (2), we introduce variables $z_{\tau i}$, equal to 1 if a TW $\tau \in \mathcal{T}$ is assigned to customer $i \in N_O$. The problem becomes a TOP with Mandatory Visits and Multiple TWs (Lin and Vincent 2017) and can be modelled as follows

$$\max \sum_{i\in N_O} y_i - \beta \sum_{(i,j)\in A} t_{ij} v_{ij} \tag{A2a}$$

$$\text{s.t.} \sum_{(i,j)\in A} v_{ij} = \sum_{(j,i)\in A} v_{ji} = y_i \qquad \forall i \in N_O \qquad \text{(A2b)}$$

$$\sum_{(i,j)\in A} v_{ij} = \sum_{(j,i)\in A} v_{ji} = 1 \qquad \forall i \in N_M \qquad \text{(A2c)}$$

$$\sum_{(0,j)\in A} v_{0j} = \sum_{(i,0)\in A} v_{i0} \le m \qquad \text{(A2d)}$$

$$a_i + t_{ij} \le a_j + H_f(1 - v_{ij}) \qquad \forall (i,j) \in A : i,j \in N \qquad \text{(A2e)}$$

$$\max\{t_{0i}, e_{\mathcal{W}_{ik}}\} \le a_i \le \min\{H_f - t_{i0}, \ell_{\mathcal{W}_{ik}}\} \qquad \forall i \in N_M \qquad \text{(A2f)}$$

$$\sum_{\tau\in\mathcal{T}} \max\{e_\tau, t_{0,i}\} z_{\tau i} \le a_i \le \sum_{\tau\in\mathcal{T}} \min\{\ell_\tau, H_f - t_{i0},\} z_{\tau i} \qquad \forall i \in N_O \qquad \text{(A2g)}$$

$$\sum_{\tau\in\mathcal{T}} z_{\tau i} = y_i \qquad \forall i \in N_O \qquad \text{(A2h)}$$

$$v_{ij} \in \{0,1\} \qquad \forall (i,j) \in A \qquad \text{(A2i)}$$

$$y_i \in \{0,1\} \qquad \forall i \in N_O \qquad \text{(A2j)}$$

$$a_i \in \mathbb{R}_+ \qquad \forall i \in N \qquad \text{(A2k)}$$

$$z_{\tau i} \in \{0,1\} \qquad \forall i \in N_O \quad \forall \tau \in \mathcal{T} \qquad \text{(A2l)}$$

Objective function (A2a) and constraints (A2b)– (A2f) are identical to objective (2a) and constraints (2b)–(2f), respectively. Constraints (A2g) link the TW assignment and arrival time at the optional customers. Constraints (A2h) ensure that if a mandatory customer is served, a single TW must be assigned and (A2i)–(A2l) define the domain of the decision variables.

Appendix A.2: Detailed Results

In this section, we present detailed results on the value of the tie-breaker and the runtime of our approach. Table A1 shows the objective value with and without routing tie-breaker. We observe that for instance with a small spread and instances with two vehicles, the differences between SL with the tie-breaker and without the tie-breaker are particularly large. In these instances, comparably many resources are available. This increases the likelihood that the upper bound relaxation leads to service for all sampled customers, especially in early states, where no real customers with TW restrict the vehicles. Notably, for the standard bound policy SL-SB, the tie-breaker is not advantageous as shown on the right side of the table.

Table A2 depicts the computation time of our policy SL with and without the tie-breaker. More specifically, for every state, we calculate the runtime required to obtain a decision (with parallelization). We then calculate the average over all states. We observe that for five of the six instance settings, the average runtime per state is below one minute. Only for the case with one vehicle and 30 expected requests, the runtime is higher, i.e., nearly three minutes. In that case, the routing becomes more challenging. This observation is confirmed by the much smaller runtimes for SL without the routing tie-breaker shown on the right side of the table.

Table A1 Value of the tie-breaker.

		SL		SL - SB	
Num. Veh./Req.	Spread	Tie-breaker	No Tie-breaker	Tie-breaker	No Tie-breaker
1/30	large	78.9%	79.0%	46.0%	46.2%
	regular	90.5%	90.2%	62.1%	62.8%
	small	96.0%	93.8%	82.7%	83.8%
2/50	large	84.6%	81.4%	60.6%	43.3%
	regular	95.8%	80.2%	57.7%	64.3%
	small	99.8%	79.9%	69.5%	52.1%

Table A2 Average decision time of parallelized SL (in seconds).

		SL	
Num. Veh./Req.	Spread	Tie-breaker	No Tie-Breaker
1/30	large	5.7	5.0
	regular	18.6	11.4
	small	166.1	15.0
2/50	large	29.9	21.6
	regular	45.1	25.9
	small	59.6	16.9

Otto von Guericke University Magdeburg
Faculty of Economics and Management
P.O. Box 4120 | 39016 Magdeburg | Germany

Tel.: +49 (0) 3 91 / 67-1 85 84
Fax: +49 (0) 3 91 / 67-1 21 20

www.fww.ovgu.de/femm

ISSN 1615-4274