The online steady-state technician booking problem

Edison Avraham, Tal Raviv
Department of Industrial Engineering, Tel Aviv University

March 2021

Abstract

We present a model for policy optimization of the online booking of mobile personnel over a multiday horizon with a different cutoff for each day, where the goal is to maximize the expected ratio of accepted requests in the long term. The interactions with the customers are performed in a single step: The system offers an assortment of time slots covering the next few days, and the user either chooses one of them or abandons the system. The model maintains a tentative routing and scheduling solution that is updated after the acceptance of each request. Upon the arrival of a service request, the provider estimates the opportunity cost of serving the request at each of the available time slots. We model this cost as a Cobb–Douglas function of features that concisely represent the current system state. The assortment of time slots following each service request is constructed by maximizing the expected net gain from the assortment. The parameters of the Cobb–Douglas function are fitted using a simulation framework. The proposed method is benchmarked based on randomly generated datasets in various demand scenarios and geographies. The method is shown to significantly outperform a more straightforward baseline policy that is commonly used as well as a more conventional linear model for estimating the opportunity cost.

Keywords: Vehicle Routing, Field Service Management, Multiday Booking

1 Introduction

Many companies provide services at their customers' locations, e.g., maintenance and repair of home appliances, grocery delivery, and medical care. Customers call the contact center of the providers or log in to their websites to book a service. In a typical situation, each customer is offered a set of possible time slots in which the service can be provided and may choose one or abandon the system. The actual scheduling and routing of the service personnel are determined before the beginning of each working day based on the service requests that should be served on that day. Requests may be booked a few days in advance, and therefore, the demand for time slots on different days is interrelated. That is, service on a particular day is a substitute for service on other days. We use the terms *period* and *day* interchangeably.

The first stage of the service provision process involves an interaction between the service provider and customer regarding the time slot in which the service should begin. For example, if the time slot agreed upon is 8:00-10:00 tomorrow, the service personnel would arrive at the customer location at any time during that slot. The provider should take into account spatial and temporal considerations when agreeing to certain time slots, while the customer has her considerations that are unknown to the provider. In this work, we refer to this interaction as the *booking process*. From the provider's side, this process is, in many cases, fully automated.

Online technician booking problems are characterized by the arrival of requests at different points in time according to some stochastic process and during a *booking horizon*. Each request is accepted or rejected *online* upon arrival, following a short interaction between the customer and the service provider. The booking horizon ends at a certain *cutoff* time that precedes the *service horizon*, during which the requests are served. Deciding whether to accept or reject a request may be done based solely on previously accepted requests, that is, *myopically*. In a more farsighted approach, the decision process may incorporate some information related to future requests; this framework is referred to as *anticipatory*.

This paper is the first to study a multiday booking problem in which requests are accepted several working days in advance, where each day has its own cutoff time. The service horizon is rolled forward by one day at the beginning of each working day. In contrast to the single cutoff case, in the multiday setting, there are inherent interactions between the various days. These interactions require optimizing the steady-state performance of the system rather than optimizing each working day separately. This paper introduces an effective policy optimization method and performs a numerical experiment that demonstrates its merits. Moreover, we demonstrate that the additional flexibility obtained from the multiday setting results in more efficient operation of the service crew.

We define the online steady-state technician booking problem (O-SSTBP) as follows: the provider is the leader in the booking process. That is, she offers several possible time slots to the customer. The offered time slots span several subsequent working days. The customer decides whether to accept one of the offered slots and place a service request or to abandon the system altogether. If the customer selects one of the offered slots, the provider is committed to serving her within the slot. To meet this commitment, the provider can offer only slots that allow serving the current customer without violating her commitments to previously booked customers. We refer to these slots as *feasible slots*.

The customers have a predefined preference over the slots. This preference consists of two components: a component common to all the customers, which is known to the provider, and an individual random component. We use a discrete choice model to describe the customer decision. The multinomial logit (MNL) model is used for demonstration in our numerical study. Similarly, the service time required by all customers is assumed to be identical. The customer uniformity assumptions can be justified because, during the booking process, the provider possesses relatively little information about the customer and the required service.

A solution to the O-SSTBP is a *policy* by which the provider constructs an assortment of time slots upon the arrival of each request. The service provider's objective is to maximize the ratio of accepted requests in the long run. We note that even though different customers may represent different values for the provider, this value may be unknown at the time of booking.

During the booking process, the service provider should consider the following trade-off. By offering many possible time slots, the provider can increase the customer's probability of accepting one of them but at the cost of a greater chance that the customer selects a slot that requires a greater travel time. Consequently, the availability of the service provider for future customer requests is reduced. Note that we do not explicitly include the travel time of mobile personnel in the objective function. However, a policy that maximizes the number of served customers within the predefined time slots implicitly minimizes the travel time.

We use the term technician to describe the mobile employee that visits customers and provides the service. However, the O-SSTBP arises in many practical situations where service requests can be placed a few days in advance. The provider cannot control the demand by dynamically setting the service fees but can control the availability of the time slots for service.

Our solution method is based on the following idea. For each time slot, we estimate the *opportunity cost* of serving the current customer in terms of the expected number of customers that will be lost if the current customer accepts the service offer. This estimate is obtained by fitting the parameters of a Cobb–Douglas function with the following arguments: (1) the remaining idle time in the slot of the technician after the customer request is inserted in the slot; (2) the remaining idle time in the daily route of the technician after the customer request is inserted in the slot; and (3) the additional travel time required to serve the request. The advantage of using a Cobb–Douglas function to represent the opportunity cost lies in the fact that it allows capturing the interactions between the above measures. For example, a large amount of remaining time in the day is not enough to indicate a low opportunity cost of a slot if it is fully utilized.

The expected net gain when an offered slot is accepted by a customer is one minus its estimated opportunity cost. The assortment of slots offered by the provider is a subset of the feasible slots, which maximizes the expected net gain based on the customers' discrete choice model. We use a simulation to estimate the system performance with the estimated opportunity cost function and iteratively fit its parameters.

The optimal assortment can be a strict subset of the feasible slots since there is an intricate interplay between service quality and efficiency. The fact that the provider withholds some of the feasible slots from customers may be perceived as lowering the service level. However, in the long run, it may give the provider additional flexibility that allows her to improve the service level by rejecting fewer customers. When bookings are allowed for several days in advance, postponing a service to one of the following days can further increase the acceptance rate of requests. Finally, we observe that the Cobb–Douglas function can approximate the opportunity cost better than a linear function used in previous studies. These insights are well demonstrated in our numerical study.

Section 2 provides a review of the state of the art in the domain of the online booking of mobile personnel and highlights the unique contribution of this study. In Section 3, we formulate the problem. Section 4 describes the proposed booking policy optimization method. Section 5 presents a numerical experiment that compares the performances of the optimized policy to other policies and explores the advantages of the multiday compared to single cutoff settings. Section 6 presents the final remarks and offers directions for future research.

2 Literature review

This paper is the first to study a technician booking problem in a multiperiod context. However, there are closely related previous studies on single-period booking and dynamic routing (both single and multiperiod). In this section, we review the existing literature and compare each of the papers relevant to the current study along the following seven dimensions:

1. The leader in the interaction between the customer and the provider. Some studies assume that the customer requests one or more particular time slots and then the provider either selects one of them or rejects the request. Other studies assume that the provider presents options to the customer and then the customer selects one of them or abandons.
2. The characteristics of the process that governs customer selection when faced with several choices (e.g., several possible time slots), is the *user choice model*. The papers reviewed below differ by their assumptions regarding this model and the knowledge of the service provider about it.
3. The *demand control method* is how the provider manipulates the requests. For the case where the provider is the leader in the negotiation processes, the literature presents two methods, either by *pricing*, i.e., setting a price for each slot, or by *availability control*, i.e., offering only a subset of the feasible slots. If the customer is the leader, the provider controls the demand by selecting one of the options offered by the customers or rejecting the request. For reviews of choice models and demand control in booking systems, see Strauss et al. (2018) and Klein et al. (2020).
4. The decision made by the provider at the time of booking may be based on a *tentative detailed routing* plan or by more abstract considerations that are based on the capacity at each service zone.
5. In practice, service providers accept booking requests several periods (typically days) in advance, and their objective is to optimize performance in the long run. Most of the existing literature focuses on a single period. Few papers look at a small finite set of service days. This study is the first to optimize the steady-state performance of the system in a setting where booking is allowed for several periods in parallel. We note that in the single-period case, optimizing each period separately results in optimizing the steady-state performances of the system, but this is not true for the multiperiod case. Hence, we compare the previous studies in terms of their approach to *periodicity.*
6. The existing literature can be classified by the farsightedness of the models presented. Some models do not use information about future requests, while others make use of forecasted information.
7. A variety of solution methods have been presented to solve booking problems. The methodology is highly related to the modeling assumptions, yet some families of solution methods are prominent.

Campbell and Savelsbergh (2005) studied a problem in which a set of potential delivery requests is known at the beginning of the booking horizon. Each request is characterized by its location, possible time slots, revenue, and probability that it will materialize. If the request materializes, the provider selects one of the time slots or rejects the request. The objective function is to maximize the total revenue from serving requests minus the routing costs. The authors presented various myopic and anticipatory decision criteria for the acceptance of newly arrived delivery requests. A GRASP heuristic solves the tentative routing problem, and the marginal profit of accepting each request at each slot is estimated and used for the booking decision. The numerical experiments demonstrated the merit of the anticipatory approach. Our study differs and extends the contribution of this work in the dimensions of negotiation leadership, the customer choice model, the demand control method, the periodicity, and the solution method.

Campbell and Savelsbergh (2006) were the first to study an online booking problem where the provider presents a different price (or discount) dynamically for each slot. A linear function that maps the slot prices to selection probabilities is known. The objective is to maximize the total net revenue after subtracting the routing costs. A GRASP heuristic solves the tentative routing problem. Upon the arrival of a request, myopically optimal pricing is obtained by a solution of a linear program. The superiority of the proposed dynamic pricing policy over more straightforward policies is demonstrated. Our study differs and extends the contribution of this work in the dimensions of the customer choice model, the demand control method, the periodicity, farsightedness, and the solution method.

Ehmke and Campbell (2014) studied an online booking problem in which each delivery request is associated with two possible time slots from a predefined set. The customer presents these slots to the provider sequentially, and the provider selects the first feasible slot with respect to her previously committed requests. The objective function is to maximize the number of accepted requests. The authors presented several possible acceptance criteria. These criteria differ from each other in the extent to which time dependency and stochasticity in travel times are considered. The underlying routing problem is solved using a scheme inspired by Solomon (1987) adapted to a time-dependent setting. The numerical experimental results indicated that the criteria that consider the time dependency and stochasticity yield routing plans that minimize the actual lateness at the customers' locations when the travel times are time-dependent and stochastic. Our study differs and extends the contribution of this work in the dimensions of negotiation leadership, the customer choice model, the demand control method, the periodicity, farsightedness, and the solution method. However, we assume that the travel times are fixed and deterministic.

Köhler et al. (2020) studied an online booking problem with availability control where the service provider can present a customer with a set of feasible time slots with two possible lengths. A specific (axiomatic) user choice model that favors shorter time slots is assumed. The objective is to maximize the number of accepted customers. The study presents decision criteria for the construction of the assortment. These criteria are related to the fleet's current utilization and the location of the delivery request. Routing is carried out based on insertion heuristics. The authors demonstrated the merits of these criteria over a baseline policy that offers short time windows only. Our study differs and extends the contribution of this work in the dimensions of the customer choice model, the periodicity, and the solution method.

Asdemir et al. (2009) studied an online booking problem where each request is associated with a certain revenue and capacity consumption. The provider presents the customer with a set of time slots that do not violate the capacity constraint and determines a price for each of the presented slots. The customer chooses one of the proposed time slots or abandons the process according to an MNL discrete choice model. The utility in the MNL model is determined by the presented price and each slot's particular characteristic. The authors devised a policy that determines the prices to maximize the expected total profit of the provider. The problem was formulated as a Markov decision process (MDP) that identifies the *opportunity cost* of serving a request in each slot, i.e., the future profit obtained if the newly arrived request is not served minus the future profit obtained if it is. The intricate interrelations among the remaining time until the cutoff, prices, and opportunity costs were examined in this work. However, the routing aspects were not considered. Our study differs and extends the contribution of this work in the dimensions of the demand control method, detailed routing, the periodicity, and the solution method.

Yang et al. (2016) studied a problem similar to that of Asdemir et al. (2009) but in a more general setting, where each request is associated with a location and the routing is considered. The price determination problem is formulated as a stochastic dynamic program. Two approximation heuristics for the opportunity cost were devised: (1) a myopic approach based on the immediate marginal cost and (2) an anticipatory approach in which the opportunity cost is obtained as a dynamic weighted average of the immediate marginal cost and an estimation of the marginal cost that is based on historical final routing plans. Unlike Asdemir et al. (2009), in this paper, a detailed tentative routing is constructed upon the arrival of each request. The routing problem is solved for each slot using a heuristic method that Yang et al. (2016) devised, inspired by Campbell and Savelsbergh (2006). The opportunity cost of each slot is derived from the routing solutions. The pricing problem is formulated as a nonlinear mathematical program and solved using a standard Newton root search method. Their numerical experiment demonstrated the merits of dynamic pricing over static pricing and the superiority of the anticipatory approximation method over the myopic approach. Our study differs and extends the contribution of this work in the dimensions of the demand control method, the periodicity, and the solution method.

Yang and Strauss (2017) solved a similar problem to that solved in Yang et al. (2016), implementing an offline approximate dynamic programming (ADP) framework using linear value function approximation (VFA) to estimate the value and calculate the prices. The delivery region is decomposed into smaller zones to make the underlying routing problem more tractable. Each zone is served by a single vehicle, and the travel distance is approximated as a linear function of the number of accepted delivery requests associated with the zone. Our study differs and extends the contribution of this work in the dimensions of the demand control method, detailed routing, the periodicity, and the solution method.

Koch and Klein (2020) studied an online home delivery booking problem with dynamic pricing, in which customers follow a choice model that generalizes the MNL. The objective function is to maximize the sum of the gross profits and the fees for the deliveries minus the routing cost. The opportunity cost of each slot is estimated using a VFA technique. Routing is done using insertion heuristics. Dummy customers, representing future expected customers, are initially included in the routing plan for a more accurate estimation of the opportunity cost. The numerical experiments demonstrated the superiority of the presented VFA estimations of the opportunity cost compared with simpler baseline pricing policies and the merits of anticipating future requests using dummy customers. Our study differs and extends the contribution of this work in the dimensions of the demand control method, the periodicity, and the solution method.

Strauss et al. (2020) studied an online booking problem in which customers can express their possible degree of flexibility regarding the desired service time. The customer choice model is the nested MNL. Upon the arrival of a request, a customer is presented with a list of bundles of time slots, and each bundle is associated with a fee. Each bundle may contain one or several slots. The customer may choose one of the offered bundles or abandon the system. The service will be provided at one of the slots in the selected bundle at the firm's discretion and without a commitment to the customer in advance. The problem is to dynamically price the bundles with the objective of maximizing the expected gross profit plus the fees minus the routing costs. Large bundles tend to be cheaper than those with fewer slots. The problem is modeled as a dynamic program. The opportunity costs of the bundles are approximated using a novel linear programming model that estimates the cost of serving each incoming delivery request. The numerical experiments demonstrated that letting customers select bundles with more than one time slot significantly improves the firm's profit. Our study differs and extends the contribution of this work in the dimensions of the demand control method, the periodicity, and the solution method.

Mackert (2019) studied an e-grocer delivery booking problem with availability control. The customers are offered a list of delivery slots to choose from with the goal of maximizing the retailer profit net of the delivery cost by selecting the best assortment of time slots. The decisions made by customers (choose one of the slots or abandon the system) are modeled by the generalized attraction model (GAM) proposed by Gallego et al. (2015). The author advocates the GAM instead of its more commonly used special case MNL since it allows circumventing an overestimation of the demand by reflecting customers' dissatisfaction with small assortments of offered slots. The approximation of the opportunity cost of each request is inspired by a method presented by Klein et al. (2018). A linear program is solved to find an optimal assortment of time slots to offer. Mackert (2019) studies a model that is similar to ours and differs only in the periodicity and solution method. Mackert (2019) considers booking for a single period, while in the current study, the service horizon consists of several periods, and the goal is to maximize the ratio of accepted requests in the long run.

Ulmer and Thomas (2019) studied an online booking problem in which each delivery request is associated with the potential revenue, location, and vehicle capacity consumption. A request is either served at any time in the service horizon or rejected. The objective function is to maximize the sum of the revenues. Routing is done using an insertion heuristic. The authors apply an ADP framework and devise a new VFA procedure that combines parametric and nonparametric VFA methods. Our study differs and extends the contribution of this work in the dimensions of the customer choice model, the periodicity, and the solution method.

With respect to the literature on booking problems, we extend the state of the art mainly in terms of the periodicity and the steady-state performance objective. Following many previous studies, we use a general user choice model where the service provider is the leader in the negotiation process. The demand is manipulated using availability control, and local decisions incorporate farsighted considerations by estimating the opportunity cost of each offered slot. Unlike previous studies, we use a nonlinear function to approximate this cost and show its merits compared to commonly used alternatives.

The much more developed literature on the dynamic vehicle routing problem (VRP) with stochastic requests shares some properties with the booking literature. In both types of problems, a customer calls a provider to set up a visit from mobile personnel. However, in the dynamic VRP setting, there is no notion of a booking horizon or cutoff time. That is, the booking and service horizon overlap. For comprehensive surveys on single-day dynamic vehicle routing problems, readers may refer to Pillac et al. (2013) and Psaraftis et al. (2016). For studies on dynamic routing with stochastic requests, see Bent and Van Hentenryck (2004), Ghiani et al. (2009), Ghiani et al. (2012), Soeffker et al. (2017), Ulmer (2019), and Voccia et al. (2019).

In most of the existing literature on the dynamic VRP, the customer presents a request that the provider can accept or reject. An exception to this modeling assumption can be found in Ulmer (2020), who studied a dynamic VRP where delivery requests are to be served on their arrival day (same-day delivery). The customer can select the delivery deadline, and the service price is determined based on his selection. That is, expedited service is more expensive. The provider assumes a willingness-to-pay choice model. The problem is to dynamically price the deliveries to maximize the expected revenue. The problem is formulated as an MDP, and a VFA scheme is used to estimate the opportunity costs of each delivery option. Routing is carried out using an insertion heuristic. The numerical experiments showed the superiority of the presented framework over fixed pricing and other benchmark policies. While Ulmer (2020) shares many similarities with the booking literature, it differs from our study in several of the dimensions mentioned above. In particular, it optimizes the revenue during a finite planning horizon rather than the steady-state performance of the system.

In most of the literature on the multiday dynamic VRP, customers place requests, and the provider serves them at some time during the service horizon directly or by outsourcing at a higher cost. The objective is typically to minimize the cost or maximize the requests served on their arrival day within a finite planning horizon. For some papers in this stream, see Angelelli et al. (2009), Wen et al. (2010), Albareda-Sambola et al. (2014), and Ulmer et al. (2018b). Pérez Rivera and Mes (2017) studied a multiperiod routing problem in the domain of freight transportation. Chen et al. (2017) solved a multiperiod technician scheduling problem where the service time of each technician is reduced as a result of increasing previous experience with similar tasks.

While some of the studies on the multiday dynamic VRP use methodologies similar to ours, their modeling assumptions are different. They assume that the customer is the leader and hence make no use of a choice model and do not optimize assortments or prices of the slots. Moreover, these papers focus on a finite horizon and thus do not optimize the steady-state performance.

Finally, several authors have studied tactical planning in the context of online booking, i.e., the offline process of determining service zones, the allocation of personnel to zones, and service time slots. See, for example, Agatz et al. (2011), Cleophas and Ehmke (2014), Hernandez et al. (2017), Bühler et al. (2016), Restrepo et al. (2019), and Klein et al. (2019).

3 Problem definition

The online steady-state technician booking problem is stated as follows: service requests arrive according to a nonhomogeneous periodic Poisson process. Each request is characterized by its location. The service time of all the requests is assumed to be identical and denoted by τ. The horizon is divided into working days, and each day is divided into several nonoverlapping service windows (e.g., 8–10, 10–12, 12–14, 14–16, 16–18). A combination of a service day and a window is referred to as a *slot*. Upon the arrival of a request on day h, it can be either rejected or scheduled for service in a slot on one of the following days $\{h+1, h+2, .., h+H^p\}$; this set of days is referred to as the *service horizon*. From the description above, it follows that a cutoff time associated with day $h+1$ is the end of day h.

The accepted requests are served by a crew of M homogenous mobile technicians available on each working day. The technicians depart from a depot, not before a predefined time, and return to the depot at most L time units later. The travel time between any two locations j and k in the service area, t_{jk}, is known and deterministic.

Each newly arrived service request is initiated by a customer and triggers the execution of the following protocol.

- The service provider's system displays an assortment of time slots in the service horizon for the service's possible starting times.
- The customer may select one of the proposed time slots or abort the process without booking his request. The customer's decision is modeled using a discrete choice model, and the parameters of this model are assumed to be known to the service provider.
- Once the customer selects a slot, the service request is booked in the system and inserted into the selected time slot's tentative schedule.

The assortment is a subset of the *feasible* slots in which the request may begin. A slot is considered feasible if assigning the request to that slot causes no late arrivals at the locations of the previously accepted service requests and allows the technicians to return to the depot by the end of the working day. The assortment may be empty, in which case the request is rejected immediately.

It should be noted that the actual schedules and routes of the technicians for each working day can be reoptimized after the cutoff time by solving a static vehicle routing problem with time windows (VRP-TW). Moreover, in a typical situation, there are several hours between the cutoff time (e.g., at midnight) and the beginning of the working day (e.g., at 7 am). During these hours, the system can exert additional computational resources to achieve a plan with a shorter total travel time compared to the original tentative plan. We note that there is vast literature on static vehicle routing problems with time windows, which is out of the scope of this study. We refer the reader to a recent survey on this topic (Braekers et al., 2016) and a paper on stochastic routing and scheduling with time slots (Avraham and Raviv, 2020).

A solution for the O-SSTBP is a policy for creating an assortment for each customer request online with the objective of maximizing the expected acceptance rate of requests in the steady state.

The description of a discrete choice model is included in the input of the O-SSTBP. We assume that all the customers follow the same choice model. In the numerical section of this paper, we use an MNL model as an example; however, any discrete choice model could be used. An advantage of the MNL model is that it is possible to estimate the parameters of the model from historical data; see Yang et al. (2016).

We formulate the problem of maximizing the expected number of accepted requests per day in the steady state as an MDP. To this end, we define the following notation:

$\mathcal{A}$	The set of time slots in the service horizon. Each subset of slots (an assortment) can be a feasible action in a particular state.
w_0	The alternative of abandoning the system.

$\mathcal{L}$	Set of locations of all potential customers. For simplicity of presentation, we assume here that this is a discrete set.
T	The number of time intervals during a single day. Each interval is short enough to neglect the probability that the number of requests arriving during the interval is greater than 1. The intervals are indexed as $t = 1, \dots, T$.
λ_t	The arrival probability of a request during interval t.
p_{lt}	The probability that a request from location $l \in \mathcal{L}$ arrives at time interval $t \in \{1, \dots, T\}$. Note that $\sum_{l \in \mathcal{L}} p_{lt} = \lambda_t$.
$\mathcal{P}(A, w)$	The probability that a customer will choose time slot w when presented with an assortment of time slots $A \subseteq \mathcal{A}$. Recall that the alternative of abandoning the system is denoted as w_0. Therefore, $\mathcal{P}(A, w_0)$ denotes the probability of not selecting any slot when the assortment A is presented. We use the convention that $\mathcal{P}(\emptyset, w_0) = 1$. Note that $\mathcal{P}(A, w) = 0$ if $w \notin A$.
S_{ht}	The state of the system at time interval t on working day h. A state includes information about the number of accepted customer requests at each particular time slot in the service horizon, $w \in \mathcal{A}$, at each location, $l \in \mathcal{L}$, and at the current time of the day t. The state can be described as a tuple of the current time and a sparse matrix with one column for each possible location and one row for each time slot. The entries of the matrix are the number of booked customers at the corresponding slot and location.
$\mathcal{S}$	The set of all feasible states.
$\pi(S, l)$	A policy π is a mapping of $\pi: (S, l) \to 2^{\mathcal{A}}$. The policy prescribes a set of feasible slots to offer to a customer from location l that places a service request while the state of the system is S. $2^{\mathcal{A}}$ denotes the collection of all subsets of time slots in the service horizon. We use the convention that $\pi(S, l_0) = \emptyset$ to refer to situations where no request arrives.
Π	The set of all policies.
$N(S_{ht})$	The set of states that can be reached from the current state in a one-time interval resulting from booking at most one customer request. For the last interval in a day, $N(S_{hT})$ consists of states $S_{h+1,1}$. Note that in the transition from time interval T to time interval 1, the cutoff time is crossed. After the cutoff time, the bookings of the first day of the previous service horizon are removed, the rest of the bookings are moved to one day earlier, and new empty slots for the last day of the new service horizon are added.
$I(S_{ht})$	For $t < T$, the state at time interval $t + 1$ if no request was accepted at time t. For the last interval $I\left(S_{h+1,T}\right)$ is the state $S_{h+1,1}$ if no request was accepted at time T or a request was accepted for the first day of the service horizon.
$\Delta\left(S_{ht}, S_{h,t+1}\right)$	For $S_{h,t+1} \in N(S_{ht})$ and $t < T$, the location of the arriving request at time t. For $S_{\mathrm{h+1,1}} \in N(S_{hT})$, the location of the arriving request at time T.
$w(S_{ht}, S_{h,t+1})$	For $S_{h,t+1} \in N(S_{h,t})$ and $t < T$, the slot of the arriving request at time t. For $S_{\mathrm{h+1,1}} \in N(S_{h,T})$, the slot of the arriving request at time T.
W_1	The set of slots on the first day of the service horizon.

Using this notation, we can define the transition probability between any pair of states in consecutive time intervals given a request at location l for $t = 1, \dots, T-1$, as follows.

$$P(S_{ht}, S_{h,t+1}) = \begin{cases} 1 - \lambda_t + \sum_{l \in \mathcal{L}} p_{lt} \mathcal{P}(\pi(S_{ht}, l), w_0), & S_{h,t+1} = I(S_{ht}) \\ p_{\Delta(S_{ht}, S_{h,t+1}),t} \mathcal{P}\big(\pi\big(S_{ht}, \Delta(S_{ht}, S_{h,t+1})\big), w(S_{ht}, S_{h,t+1})\big), & S_{h,t+1} \in N(S_{ht}) \\ 0, & otherwise \end{cases} \quad (1)$$

The first case refers to transitions resulting from progressing from time interval t to $t+1$ without accepting any request. Note that $S_{h,t+1} = I(S_{ht})$ indicates that the only difference between the states S_{ht} and $S_{h,t+1}$ is the time. The probability of this transition is the sum of two probabilities of two mutually exclusive cases. Either no customer request arrives during interval t (with a probability of $1 - \lambda_t$) or a request arrives, an assortment $\pi(S_{ht}, l)$ is offered by the operator and the customer rejects it. The second case refers to transitions resulting from progressing from time interval t to $t+1$ where a request is accepted. The location of the accepted request is encoded in the information regarding the states S_{ht} and $S_{h,t+1}$ as $\Delta(S_{ht}, S_{h,t+1})$ and the slot as $w(S_{ht}, S_{h,t+1})$. The third case refers to all the other transitions, which are infeasible in a one-time interval. For time interval T, just before the cutoff time, the transition probabilities are

$$P(S_{hT}, S_{h+1,1}) = \begin{cases} 1 - \lambda_T + \sum_{l \in \mathcal{L}} p_{lT} \sum_{w \in W_1 \cup \{w_0\}} \mathcal{P}(\pi(S_{hT}, l), w), & S_{h+1,1} = I(S_{hT}) \\ p_{\Delta(S_{hT}, S_{h1}),t} \mathcal{P}(\pi(S_{ht}, \Delta(S_{hT}, S_{h1})), w(S_{hT}, S_{h1})), & S_{h+1,1} \in N(S_{hT}) \\ 0, & otherwise \end{cases} \quad (2)$$

The first case refers to transitions from time interval T to 1 without accepting any request or when the accepted request is for the first day of the service horizon and hence is deleted from the state at the cutoff time at the end of time interval T. The first term is as shown in (1), and the second term is a sum over the events where the customer either chooses to abandon the system or chooses a slot on the first day of the service horizon. The second and third cases are exactly the same as in (1), but note the unique definitions of $N(S_{hT})$, $\Delta(S_{hT}, S_{h+1,1})$, and $w(S_{hT}, S_{h+1,1})$.

When a policy π is known, the transition probability above defines a Markov chain over the states $S \in \mathcal{S}$. We denote the steady-state probability of this chain at time interval t by $Q_t(S, \pi)$.

An optimal solution is a policy that satisfies

$$\pi^* = \underset{\pi \in \Pi}{\operatorname{argmax}} \sum_{t=1}^{T} \sum_{S \in \mathcal{S}} Q_t(S, \pi) \cdot \sum_{l \in \mathcal{L}} p_{lt} \sum_{w \in \pi(S,l)} \mathcal{P}(\pi(S, l), w)$$

Recall that we are interested in the steady-state performance of the system when the demand is assumed to be stationary. Therefore, it is enough to consider the expected number of accepted requests in one typical day.

The expected number of accepted requests per day in the steady state is calculated as the sum of the probabilities of accepting a request in each time interval. For each particular state S, the probability of accepting a request in the corresponding interval is calculated by the sum of the probability of the arrival events multiplied by the probability that one of the offered slots will be accepted.

Unlike previous authors in the booking literature, we used a general MDP rather than a sequential set of Bellman equations to formally define our problem. The MDP framework is suitable because we are interested in the expected number of accepted requests in a steady state rather than the number of accepted requests within a finite booking horizon.

In practice, customers differ in their characteristics, such as expected service time and choice model. If the service provider is aware of the particular characteristics of the customer when a request is booked, then a more detailed model can be used to achieve better service. To this end, the state space may include additional properties such as the customer type or the arrival time of the request. The method for approximating the value function presented in the next section can be adapted for such a detailed state description, but this would require fitting a higher-dimensional function that may be much more difficult to estimate.

4 Methodology

The direct solution of the MDP presented in Section 3 is not practical for realistic settings since the set of states is too large. Instead, we encode each state and the information regarding each new request in a compact way and use simulation results to fit a Cobb–Douglas function that approximates the opportunity cost of accepting a request from each possible location in each time slot in the service horizon. Upon the arrival of a service request, an assortment of feasible slots that approximately maximizes the expected gain from the customer is constructed and offered to the customer. This assortment is obtained by selecting one that maximizes the immediate (unit) gain minus the approximated *opportunity cost*. At any particular moment, the booking system maintains a tentative set of routes and schedules for all technicians within the entire service horizon. Once a customer selects a time slot, his request is inserted into one of the tentative routes, and the schedule is updated. The tentative plan and the insertion procedure enable us to quickly evaluate the feasibility of inserting the request in each slot.

Our solution method can be classified as a VFA (see Powell (2019)). The opportunity cost is the expected number of customers that will be lost from consistently allocating a particular slot to a customer at a given state of the system in terms of the expected steady-state performance. In previous studies that analyzed finite horizon booking problems, the opportunity cost of a slot was regarded simply as the difference between the value of the states of the system with and without the booked customer; see, for example, Mackert (2019), Koch and Klein (2020), and Ulmer (2020). In our settings, the relation between the opportunity cost and the value of the policy is more intricate since the values of the states represent the daily rate of accepted customers. The opportunity cost can be viewed as the difference in the steady-state acceptance rate if the slot would be allocated to any identical customer that will arrive in the same state.

The idea of estimating the opportunity cost is well established in the booking literature, as discussed in Section 2. However, we present the first application of this idea for optimizing the long-term performances of a booking process rather than focusing on a single period with a single cutoff.

Section 4.1 presents the approximated opportunity cost function and explains the modeling considerations. Section 4.2 presents the routing and scheduling oracle that we implement to aid in the cost estimation for each request in each slot. Section 4.3 presents the optimization model used to construct the list of offered slots. Finally, Section 4.4 presents the proposed procedure for fitting the parameters of the opportunity cost function.

4.1 The approximated opportunity cost function

The approximated opportunity cost function represents the implications of accepting a particular customer request at a particular slot. The function estimates the number of requests that will either be rejected or abandon the system due to the resources being allocated to the current one.

Consider a route r' that represents the set of requests that a technician visits in a given day, as well as the starting times at each customer location and the departure and arrival times at the depot. Let i be a newly arrived request that is a candidate to be added to route r' in time slot s. A new route r, consisting of all the requests in r' and request i, is feasible if the slots of all the previous requests in r' are not changed and the service of request i starts in time slot s. Based on (r, r', i, s), one can calculate the following three measures:

$RTS(r,s)$	Remaining *idle* time in slot s (after the insertion of i into the route in slot s and assuming that the technician departs from the previous customer as early as possible). The technician is considered idle when no service or travel is carried out.
$RTR(r,s)$	Remaining *idle* time for route r during the entire working day (after the insertion of i into route r in slot s).
$TT(r',r)$	Increase in the travel time incurred by extending r' to r.

The calculation of RTS, RTR, and TT is discussed in detail in Section 4.2. Note that the values of the above three measures are nonnegative. Next,

$$f(r,r',s) = \begin{cases} RTS(r,s)^{\alpha} RTR(r,s)^{\beta} TT(r',r)^{\gamma}, & if\ route\ r\ is\ feasible \\ \infty, & otherwise \end{cases} \tag{3}$$

where α, β, γ are parameters that characterize the opportunity cost function f and may be either positive or negative. The Cobb–Douglas function $f(r,r',s)$ represents the expected number of future requests that will be lost if the system accepts the current request in route r in slot s. Therefore, $f(r,r',s) > 1$ implies that serving the current request in route r in slot s is not worthwhile. Economists often use the Cobb–Douglas function to represent production and utility with respect to the inputs. We use it to approximate the opportunity cost because it is one of the simplest functional forms that can capture various interactions between inputs.

The measures RTS, RTR, and TT concisely capture sufficient information about the system state to estimate the opportunity cost effectively. The case for TT is straightforward: a higher marginal increase in the travel time corresponds to a higher likelihood that future requests will be lost. Thus, γ is likely to be positive. Next, high values of RTS and RTR indicate that the candidate route (and in the case of RTS, the candidate slot as well) is sufficiently vacant to serve additional requests after accepting request i. Thus, accepting i is unlikely to adversely affect the offers made in response to future customer requests.

Hence, α and β are likely to be negative. We note that Ulmer et al. (2018a) and Koch and Klein (2020) use similar measures to aggregate the state space in their ADP frameworks. In Section 4.4, we present a method for fitting appropriate values of α, β, γ.

Note that $RTS, RTR,$ and TT may be zero. In this case, since the exponent may be negative, to ensure that the Cobb–Douglas cost function is defined, we correct (3) by adding a small $\varepsilon > 0$ to each of the measures. Specifically,

$$f(r,r',s) = \begin{cases} (RTS(r,s)+\varepsilon)^{\alpha}(RTR(r,s)+\varepsilon)^{\beta}(TT(r',r)+\varepsilon)^{\gamma}, & if\ route\ r\ is\ feasible \\ \infty, & otherwise \end{cases}. \tag{3'}$$

4.2 The routing and scheduling oracle

This section presents a procedure to insert new requests into the existing plan at particular routes and time slots. This procedure is used to construct the tentative routes and schedules and approximate the corresponding opportunity costs. These approximate values are later used to construct assortments offered to the customers, as described in Section 4.3. The opportunity cost of a time slot is defined as the opportunity cost of assigning the request to the technician that minimizes it.

In Algorithm 1, we present a pseudocode of a program that finds the set of time slots with opportunity costs smaller than 1. For each such time slot, the program returns the technician to which the request should be tentatively assigned, the tentative position of the request along his route, and the approximate opportunity cost of the time slot. It should be noted that the routing and scheduling problem is not the focus of this study and that any other fast routing procedure can replace the proposed oracle.

```
Input:
  Set of service time slots at the next H^p working days, denoted by S
  Set of technicians, denoted by K
  Tentative routes of all the technicians in the next H^p working days
  Parameters of the opportunity cost function α, β, γ
  Identification of the current request, i, and its properties

for s ∈ S
  C[s].f ← ∞
  for k ∈ K
    r' ← current route of k in the day of slot s
    for p = 0 to |r'|                    // where |r'| is the number of customers in route r'
      r ← route of k after inserting request i after position p to r'
      if route r is feasible for slot s
        calculate RTS(r,s), RTR(r,s), TT(r',r), f(r,r',s)
        if f(r,r',s) < C[s].f
          C[s].tech ← k
          C[s].f ← f(r,r',s)
```

$C[s].route \leftarrow r$
$S' = \{s \in S | C[s].f \leq 1\}$
Return: S', C

Algorithm 1: Pseudocode for finding a set of candidate slots

For each slot s, we define an object $C[s]$ that stores information regarding the best-found route to serve request i in slot s. In particular, $C[s].tech$ is the identity of the most appropriate technician, and $C[s].route$ is the route of this technician, including the newly arrived request i. Recall that a route is defined by a set of requests and their service times. In $C[s].f$, we store the approximated opportunity cost for the particular request and slot. We initialize $C[s].f$ to infinity.

Our algorithm updates C by looping through all the slots, technicians, and insertion positions along the routes. Note that inserting a request at position p means that the request is inserted immediately after the p^{th} customer in the sequence, with the convention that insertion at position 0 means inserting at the first position along the route. After the insertion, the starting times of all requests starting from the position of the inserted request are set to the earliest possible times in their pre-committed slots. Recall that the obtained route is feasible if all its requests start at their required time slots.

The technician's route and schedule with the minimal opportunity cost are found for each slot. Finally, a list with all the slots with an expected opportunity cost that is not greater than one, S', and the best-found technicians, routes, and their opportunity costs (stored in C) are returned. A subset of the slots in this list is offered to the customer, as discussed in Section 4.3.

Next, we elaborate on the calculation of RTS, RTR, and TT. To calculate TT, let j and k denote the previous and next customers of i along the new route r, respectively. Thus, k follows j along the old route r'. Now, $TT = t_{ji} + t_{ik} - t_{jk}$.

Let u_j denote the service starting time at the location of request j, and let a_s and b_s denote the starting and ending times of slot s, respectively. The routing and scheduling oracle inserts all the customers at the earliest possible starting time u_j that maintains their time slot constraints. Specifically, if customer k is the immediate successor to customer j and k is assigned to slot s, $u_k = \max(a_s, u_j + \tau + t_{jk})$, where τ is the service time. To calculate the RTS for slot s, let $\hat{j}$ denote the customer whose service starting time is the latest in slot s. That is, $\hat{j} = argmax_{j \in R}\{u_j | a_s \leq u_j \leq b_s\}$, where R is the set of customers along route r. Next, let $\hat{k}$ denote the immediate successor of $\hat{j}$. Hence,

$$RTS = b_s - \min\left(u_{\hat{j}} + \tau + t_{\hat{j}\hat{k}}, b_s\right).$$

After serving customer $\hat{j}$, the technician is ready to start the service of $\hat{k}$ at time $u_{\hat{j}} + \tau + t_{\hat{j}\hat{k}}$. If this time is still within slot s, i.e., before b_s, the remaining time, RTS, is $b_s - (u_{\hat{j}} + \tau + t_{\hat{j}\hat{k}})$; otherwise, it is zero.

Finally, the RTR is calculated as the difference between the length of the working day, L, and the total travel and service time allocated to the technician.

4.3 Constructing an assortment

After calculating the approximated opportunity cost of scheduling the request in each slot, the planner can construct an assortment of slots to offer that will approximately maximize the expected number of current and future requests that will be served. Recall that the net gain from serving a customer in a particular slot is one minus the expected number of future requests that will be lost due to the resources allocated for the current request. The problem of selecting a subset of slots from the candidate slot list, S', which is created as described in Section 4.2, to maximize the expected net gain can be formulated as a nonlinear mixed-integer program (4)–(6). For each $s \in S'$, we define a binary decision x_s that indicates whether the slot is included in the assortment and an auxiliary variable P^s that holds the probability that the customer will select slot s.

$$maximize \sum_{s \in S'} P^s (1 - C[s].f) \tag{4}$$

$subject\ to$

$$P^s = g_s(\mathbf{x}) \qquad \forall s \in S' \tag{5}$$

$$x_s \in \{0,1\} \quad \forall s \in S' \tag{6}$$

The objective function maximizes the expected net gain from the offer by weighting the expected gain from each slot with the probability that the customer will select it. Constraint (5) relates the auxiliary variables P^s to the offered slots represented by x. The function $g_s(\mathbf{x})$ represents the discrete choice model and calculates the probability of selecting slot s when the assortment is represented by the characteristic vector **x**. We also comment that

1. For a slot s that is not included in the offer (i.e., $x_s = 0$), $P^s = 0$.
2. The sum $\sum_{s \in S'} P^s \leq 1$. This inequality is strict if the probability of customer abandonment is positive.

If the choice model is MNL, (5) can be replaced with

$$P^s = \frac{e^{u_s} x_s}{\sum_{q \in S'} e^{u_q} x_q + 1} \qquad \forall s \in S' \tag{5'}$$

In the denominator, we add the term 1 to represent the abandonment alternative. Note that the utility to a customer from abandoning the system is 0, and $e^0 = 1$.

In the numerical experiment, the optimization problem (4)–(6) is linearized according to an idea presented by Mackert (2019). A commercial solver solves the resulting linear program in fractions of a second. Note that this linearization is applicable to the GAM, which is a generalization of the MNL model. However, solving the assortment model for other choice models may be NP-hard. In these cases, one may either use enumeration of all the possible assortments or use heuristic methods.

Once the optimal assortment is presented to the customer, he may either select one of the slots or abandon the system. If the customer selects a slot s, the request is inserted into the route of technician

$C[s].tech$, as prescribed by Algorithm 1. The resulting plan is tentative, and the actual routes and schedule may be reoptimized later.

4.4 Fitting parameters for the opportunity cost function

In this section, we present a procedure for fitting values for the opportunity cost model parameters. For any combination of the parameters α, β, and γ, the opportunity cost function f defines a policy with a (possibly) different expected number of accepted requests per day. The objective is to determine the model parameters that maximize this number. The procedure described below converges to a local maximum of this optimization problem, but convergence to a global optimum cannot be ensured. In Section 5, we provide numerical evidence of the effectiveness of the proposed heuristic.

The proposed procedure relies on a variant of the steepest descent method. The procedure has three phases that are applied repeatedly: 1) estimation of the steady-state performance of a current solution; 2) estimation of the directional derivatives for the three parameters; and 3) improvement of the current solution by applying a line search in the most promising direction. The process is repeated until a stopping criterion is met.

In Phase 1, we estimate the expected number of accepted requests for a given set of parameters using a discrete event simulation. Specifically, we simulate multiple working days and calculate an estimator for the expected number of accepted requests in the steady state by ignoring the first few (warmup) days. We use common random numbers to create the same stream of orders in each iteration during the optimization process. This technique reduces the variability in the simulation results. We run a relatively long single replication in each iteration. The accuracy of the simulation is analyzed and controlled by dividing the long replication into multiple shorter blocks. By applying the blocking method, we can extract multiple observations from a single replication without running the warmup period multiple times. For a further discussion of variance reduction techniques and the statistical analysis of simulation results, see, for example, Ross (2013).

In Phase 2, we estimate the expected number of accepted requests for 26 neighboring combinations of α, β, and γ. In our framework, the neighborhood of a current solution $\boldsymbol{x}^{(n)} = (\alpha^{(n)}, \beta^{(n)}, \gamma^{(n)})$, denoted by $\mathcal{N}$, is obtained as the set of all the solutions generated from increasing, fixing, or decreasing the value of the three parameters by a small nonnegative constant Δ. The neighborhood is denoted as

$$\mathcal{N}\left(\alpha^{(n)}, \beta^{(n)}, \gamma^{(n)}\right) = \{\alpha^{(n)}, \alpha^{(n)} + \Delta, \alpha^{(n)} - \Delta\} \times \{\beta^{(n)}, \beta^{(n)} + \Delta, \beta^{(n)} - \Delta\} \times \{\gamma^{(n)}, \gamma^{(n)} + \Delta, \gamma^{(n)} - \Delta\}$$

The expected number of accepted requests of each solution in $\boldsymbol{x} \in \mathcal{N}$ is evaluated by simulation.

In Phase 3, if one or more of the neighboring solutions is better than $\mathbf{x}^{(n)}$, let $\mathbf{x}^{(n')}$ be the best solution in $\mathcal{N}$ and thus $\mathbf{x}^{(n')} - \mathbf{x}^{(n)}$ be the best *improving direction.* Next, we initiate a line search along a segment of the ray from $\mathbf{x}^{(n)}$ in the direction of $\mathbf{x}^{(n')}$, that is, in

$$\mathcal{M} = \left\{(\alpha, \beta, \gamma) | \mathbf{x}^{(n)} + l\left(\mathbf{x}^{(n')} - \mathbf{x}^{(n)}\right),\ \ l \in \Re_{+}\right\}$$

We use the golden section method to find a locally optimal solution $\boldsymbol{x}^{(n+1)}$ in ray $\mathcal{M}$. Note that since $\boldsymbol{x}^{(n')} \in \mathcal{M}$, the best solution obtained by the line search is at least as good as $\boldsymbol{x}^{(n')}$ and thus strictly better than $\boldsymbol{x}^{(n)}$.

If no improving solutions are found in $\mathcal{N}$, then $\boldsymbol{x}^{(n+1)} \leftarrow \boldsymbol{x}^{(n)}$, and Δ is updated as $\Delta \leftarrow {}^{\Delta}/_{\omega}$, where $\omega > 1$ represents an update factor. Note that reducing Δ redefines the neighborhood with more similar solutions to the current one. This can be beneficial in later stages of the search where the current solution is already close to a local optimum.

The search procedure starts from some initial solution $\boldsymbol{x}^{(0)} = \left(\alpha^{(0)}, \beta^{(0)}, \gamma^{(0)}\right)$ and iterates to generate a series of solutions with nondecreasing values. The value of Δ is reset to its initial value after each improvement. If no improving solution is found for a predefined number of iterations, the search process is terminated.

5 Numerical experiments

This section presents the numerical experiments performed to benchmark the proposed method and derive some insights. Section 5.1 describes the benchmark problem instances. Section 5.2 describes the performance analysis for the proposed method and reports the results. In Section 5.3, the importance of each of the model parameters is tested. In Section 5.4, we demonstrate the merits of a long service horizon with multiple cutoff times.

5.1 Description of problem instances

This section describes the settings of the benchmark instances used in the main experiment reported in Section 5.2. Some of the parameter values are diversified in Section 5.4 to gain additional insights.

The service crew in our benchmark instances consists of $M = 6$ technicians that are available on each working day. The working day begins at 8:00 and ends at 18:00; thus, $L = 600$ minutes. Each working day is divided into five nonoverlapping service windows of two hours each, i.e., 8:00 – 10:00, 10:00 – 12:00, 12:00 – 14:00, 14:00 – 16:00, and 16:00 – 18:00. All the slots of the same working day have the same cutoff time, which is the beginning of the corresponding working day.

The length of the service horizon is three days, $H^p = 3$. Therefore, 15 potential time slots may be available for the service of a newly arrived request. The customers are assumed to follow the MNL choice model. The expected utilities of obtaining the service in each of the five slots on the next day, i.e., the first day of the service horizon, are presented in Table 1.

Table 1: Expected utility of each possible slot

	8:00 – 10:00	10:00 – 12:00	12:00 – 14:00	14:00 – 16:00	16:00 – 18:00
Next Day	3	2	1	2	3

Each additional day of delay in the service results in a 20% reduction in the utility associated with each slot. The utility obtained from abandoning the process is 0. Recall that the expected utilities are assumed to be the same for all customers. This preference model reflects the fact that the customers desire to be served as soon as possible and that the time slots at the beginning or the end of the working day are more convenient than the mid-day slots.

In our experiments, the booking system is fully automated (e.g., via a website), and the service requests may arrive at any hour during the day. The arrival process of the requests is assumed to be periodic (daily). The fraction of the arrivals during each hour of the day, denoted by d_t, is presented in

Figure 1. Note that the arrival process reflects two peaks: one in the late morning and another in the early evening. Naturally, the arrival rate is considerably low during the nighttime.

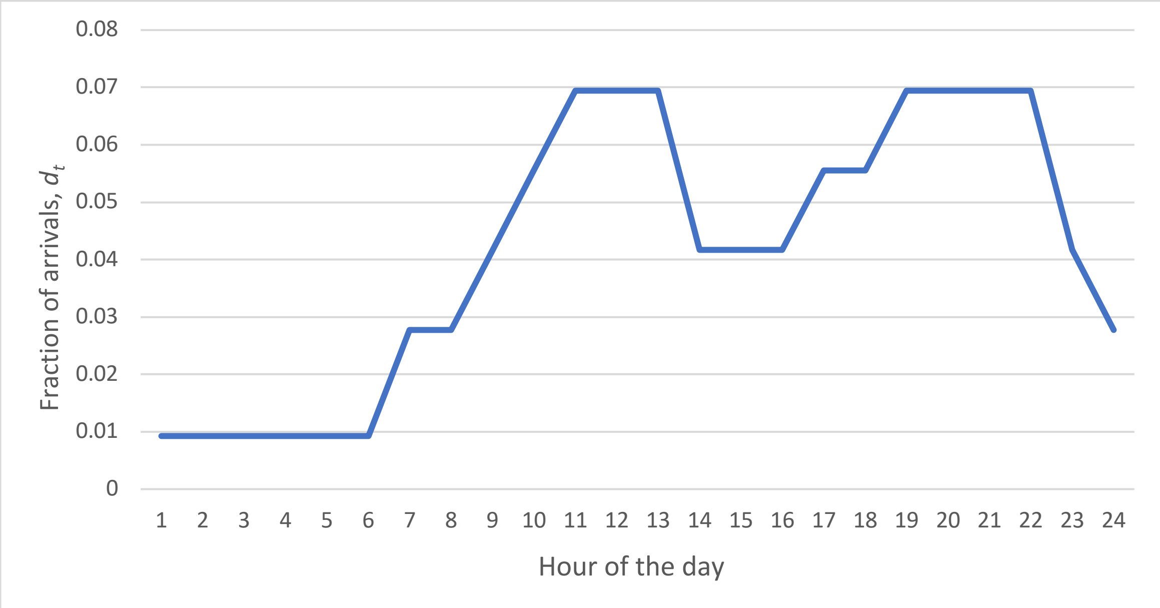


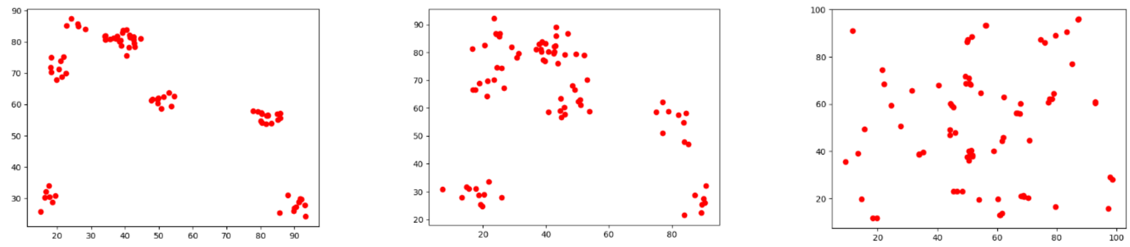
Figure 1: Fraction of requests arriving at each hour of the day

In our problem instances, we consider three values of the expected number of requests per day, denoted by D, namely, 64, 72, and 80 requests. The service time τ related to each instance is determined such that the expected amount of total required service time is identical among all the instances and equals ${}^{2}/_{3}$ of the six technicians' total working time, i.e., 2,400 minutes, which represents 400 minutes for each of the technicians. For example, for instances with 80 customers, the service time per customer is $\tau = {}^{2400}/_{80} = 30$ minutes.

In all the problem instances, the locations of the customers are scattered on a 100×100 square, and the depot is located at the center. We generate three types of spatial configurations, namely, urban, suburban, and rural. For the urban and suburban configurations, 8 points are randomly generated from a uniform distribution, i.e., $x_z \sim U(10,90), y_z \sim U(10,90), z = 1 \dots 8$. Each point constitutes the center of a zone. For the rural configuration, 72 points are generated to represent the centers of zones. In this case, $x_z \sim U(2,98), y_z \sim U(2,98), z = 1 \dots 72$. The locations of the settlement centers are kept constant throughout the experiment.

Upon the arrival of each request, its location is generated in two steps: first, we randomly pick a zone, z, for it, and then we generate its exact location using random polar coordinates. An angle θ and distance r are randomly generated using a uniform distribution $\theta \sim U(0,2\pi), r \sim U(0, R)$, where $R = 5, 10,$ and 2 for the urban, suburban, and rural configurations, respectively. Next, based on the polar coordinates, (θ, r), the x coordinate is obtained as $x = x_z + r cos(\theta)$, and the y coordinate is obtained as $y = y_z + r sin(\theta)$. The travel times between the locations are the Euclidean distances.

The urban and suburban configurations are created to reflect a setting with a small number of high-demand cities. The travel time within each city is short to moderate. The rural configuration represents a highly dispersed setting in which the expected demand at each settlement (village or small town) is low. Figure 2 demonstrates the locations of requests that arrive on a typical day for the three types of configurations.



Urban Suburban Rural

Figure 2: Example of the locations of the requests in each spatial configuration

A class of instances is determined by the spatial configuration and the demand intensity D since all the other parameters are constant. A training stream and a testing stream of requests are generated for each instance. The training stream is used to fit the parameters of each instance. The same stream is used for all the parameter combinations explored by our search procedure. The testing stream is used to evaluate the performance of the booking system with the final parameters, (α, β, γ), fitted in the search process. The training and test streams consist of requests obtained over 600 and 6000 booking days, respectively. They include 60 warmup days that have been excluded from the estimations to ensure that the observed measures represent the steady state. The lengths of the simulation and the warmup period are determined and verified by extensive statistical analysis using a blocking technique.

We generate three instances with different random zone centers and random streams of requests for each unique combination of the spatial configuration (urban, suburban, rural) and the value of D (64, 72, 80 requests per day). There are 27 benchmark instances in total. A Python script that reproduces this dataset and its detailed description are available in the electronic appendix, which can be downloaded from http://www.eng.tau.ac.il/~talraviv/Publications/.

The instances are coded according to the number of requests D, the number of zones Z, the radius of the settlements R, and the serial replication number S. For example, the code name D72Z8R10S2 refers to the second instance with 72 requests per day and eight different settlements with a radius of 10 each (suburban setting).

5.2 Performance evaluation of the proposed method

In this section, we compare the performance of the proposed method with that of several commonly used policies from the literature for generating assortments and show the superiority of our method. We aim to gain insights related to the following questions:

- How well does the proposed method perform compared to the previously proposed and used policies in terms of the acceptance rate? Maximizing the acceptance rate is the objective of the assortment optimization process. In particular, what are the merits of fitting the opportunity cost to a Cobb–Douglas function rather than the previously proposed approximation schemes?
- How does the proposed method affect the proportion of unserved customers, namely, customers who are offered no slots or customers who choose not to book a time slot even after a nonempty

assortment of possible slots is offered to them? The arriving customers are divided into three types: accepted, rejected, and abandoning. The division between the second and third types can be seen as an indicator of customer satisfaction.

- Does the proposed method induce more efficient routing plans and schedules? While the traveling distance is not explicitly included in the objective function, it is still of interest to the service provider.
- What is the typical structure of the assortments offered to customers? To answer this question, we report the cardinality and average utility of an assortment. Both are indicators of the quality of the offers. The *utility of an assortment* is defined as the sum of the utilities of the slots included in the assortment.
- What is the quality of the actual service the customers receive? To answer this question, we report the following features: the average utility of the (actual) service slot and the proportion of customers served on each day of the service horizon.

The following policies are considered:

- A myopic baseline policy, denoted as MYOPIC. Under this policy, upon the arrival of a request, all feasible slots are offered. It should be noted that under any reasonable choice model, the optimal assortment in a policy that ignores future requests includes all the feasible time slots since extending the assortment by additional slots cannot increase the probability of abandonment. In particular, in the MNL model used in our experiment, any additional slot in the assortment strictly decreases the chance that a customer will abandon the booking process. Therefore, the assortment offered by our baseline policy is optimal in the myopic sense.
- A rule-of-thumb policy, denoted as X-SLOTS. Under this policy, upon the arrival of a request, only a subset of at most X feasible slots most valuable to the customer is offered, i.e., the feasible slots with the highest expected utility. This rule of thumb is reasonable to follow when resources for intense optimization of the assortments are not available but the service provider wants to withhold some slots for future customers. In our experiment, we tested the 3-SLOTS and 5-SLOTS policies.
- A policy based on fitting the opportunity cost to a linear-in-parameters function, denoted as LINEAR. Under this policy, the approximated opportunity cost function is a linear function of the same three measures RTS, RTR, and TT rather than a Cobb–Douglas function, i.e., $f = \alpha RTS + \beta RTR + \gamma TT$ for all feasible slots, and ∞ otherwise. We note that the α, β and γ parameters are fitted using the same approach as that of our proposed policy.
- Our proposed policy, denoted as COBB-DOUGLAS.

All the policies are compared based on the same geographies and testing streams of the requests. The same customer choice model and routing oracle (see Section 4.2) are used. For the COBB-DOUGLAS and LINEAR policies, the search process begins with an initial solution: $\alpha^0 = 0, \beta^0 = 0, \gamma^0 = 0$. The initial Δ is set as 0.1, and it is updated by a factor of ω=2. The numerical tolerance of the Cobb–Douglas function (COBB-DOUGLAS policy) is $\varepsilon = 0.01$. The process stops after four consecutive iterations without improving the objective function. In the electronic appendix, we provide a table with the best-obtained values of the parameters for each instance and the two policies. In both

LINEAR and COBB-DOUGLAS, the values of γ are strictly positive, and the values of α and β are strictly negative.

The main insights are visualized below. Detailed statistics regarding the performance of each of the five policies for each of the 27 instances can be found in the electronic appendix.

Figure 3 displays the average proportion of abandoning and rejected customers for the three demand rates 64, 72, and 80 and the five policies described above. The calculations are based on the 6000-day test stream and exclude the warmup. Rejected customers are those whose assortment is empty. Customers presented with a nonempty assortment that chose to end the process without booking are considered abandoning customers. The rest of the customers are considered served.

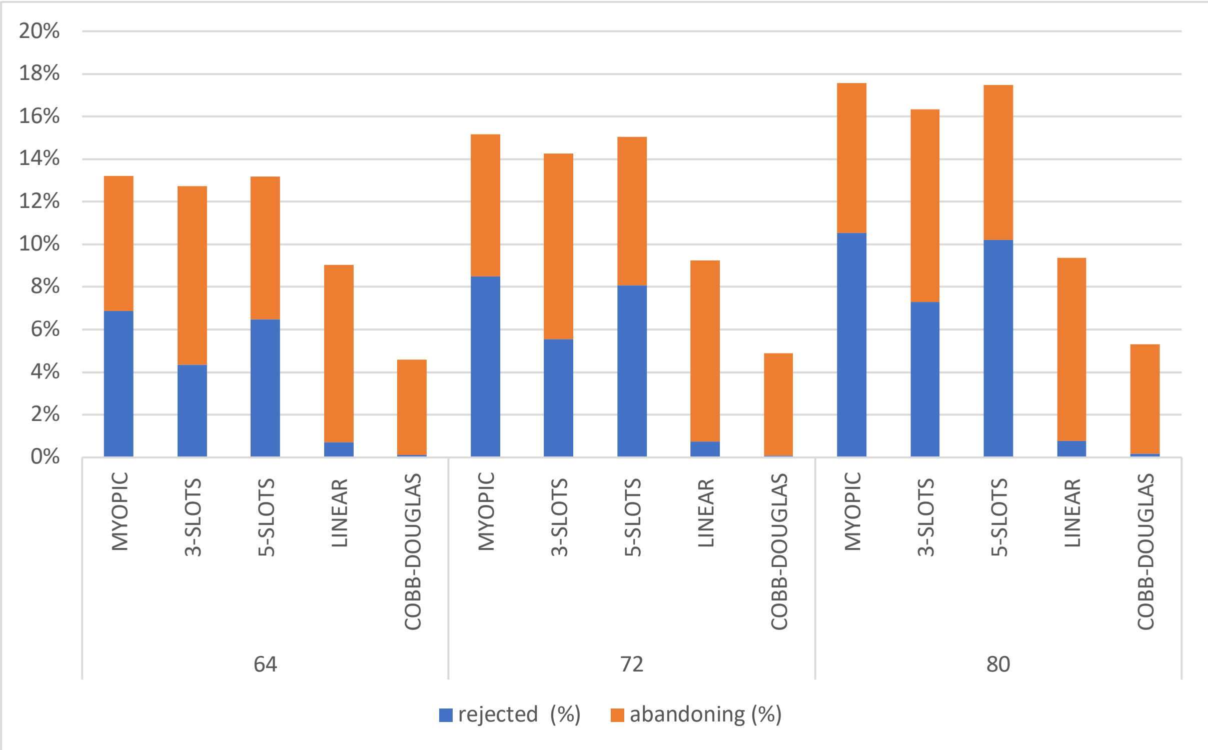


Figure 3: Average proportions of customers served, rejected, and abandoning for all policies and demand values

Figure 3 illustrates the superiority of COBB-DOUGLAS over the other policies quite clearly. For all demand rates, the proportion of served customers is the highest under this policy. As shown in the electronic appendix, COBB-DOUGLAS outperforms each of the other policies in each of the 27 instances. This result is significant (p-value < 0.001 in four paired t-tests). Moreover, the proportion of rejected customers, i.e., those with an empty assortment, is negligible. In addition, COBB-DOUGLAS has the lowest proportion of abandoning customers. In terms of the proportion of served customers, LINEAR is second best. However, note that its proportion of abandoning customers is quite high.

We also note that the rule-of-thumb policies are slightly better than MYOPIC in terms of served customers. In paired t-tests carried out using the results of the 27 instances, the acceptance rates of both 3-SLOTS and 5-SLOTS are greater than that of MYOPIC (p-value < 0.001).

Figure 4 displays the average travel time per served request for the three configurations (urban, suburban and rural) for the five tested policies.

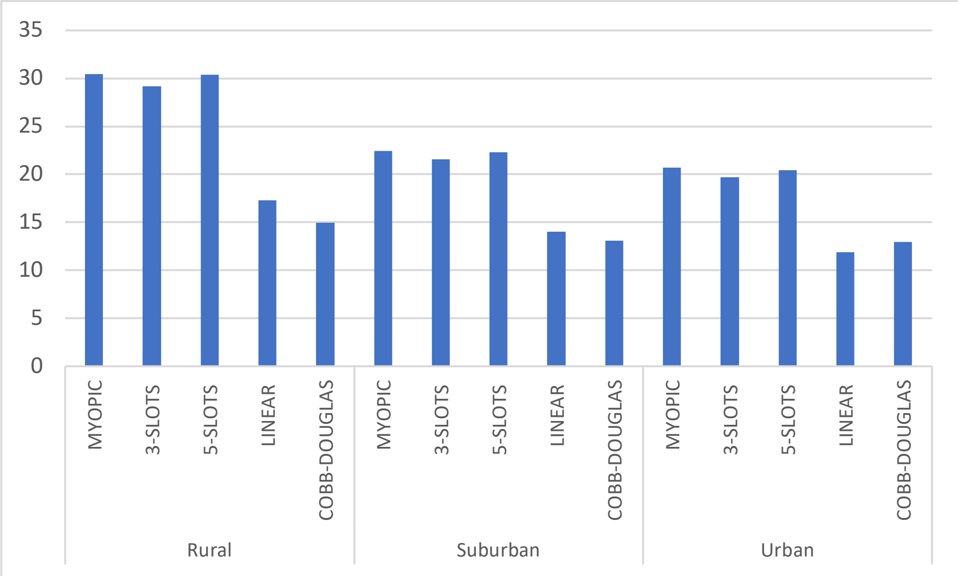


Figure 4: Average travel time per served request for the three configurations

It is clear that the two policies that apply advanced assortment optimization methods, namely, COBB-DOUGLAS and LINEAR, lead to considerably shorter travel times compared to those of the other policies and thus utilize the fleet much more efficiently. The effect is most significant for the rural instances. A decrease of almost 50% in the average travel time is noted when following COBB-DOUGLAS rather than MYOPIC or the two rule-of-thumb policies. This advantage of COBB-DOUGLAS is expected since when customers are sparser, assortment optimization can save more in terms of the travel time. COBB-DOUGLAS yields slightly lower average travel times than LINEAR in the rural and suburban configurations, while LINEAR yields a slightly lower average travel time for the urban configuration. However, note that this comes at the cost of a significantly lower acceptance rate.

Figure 5 displays the average number of slots offered on each of the three days of the service horizon for each of the five policies and three demand rates. These averages are calculated based only on customers whose offered assortments are nonempty, that is, excluding the rejected customers.

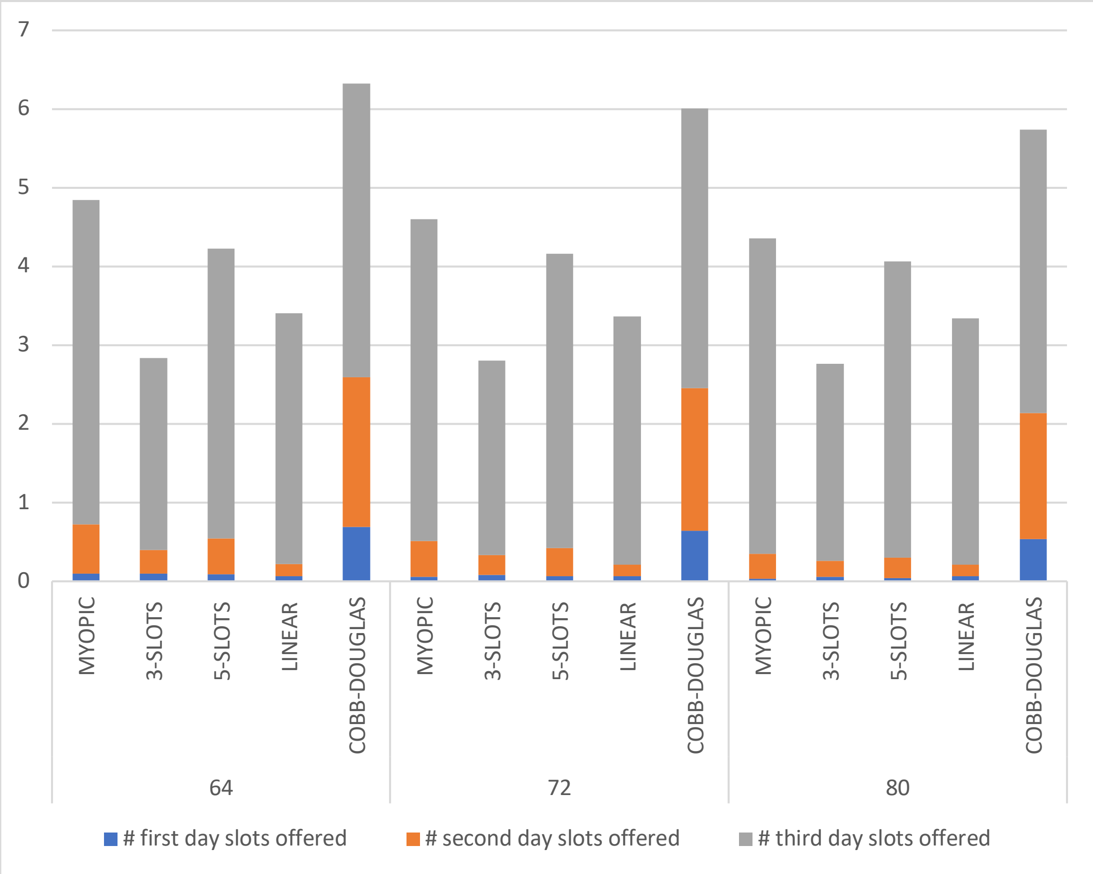


Figure 5: Number of slots offered on each working day for the different policies

It is apparent from Figure 5 that COBB-DOUGLAS is superior to the other policies in terms of both the total number of slots offered and the number of slots offered for the first and second days. COBB-DOUGLAS is the only policy for which the assortments include a nonnegligible number of first-day slots. It is interesting to note that MYOPIC is better than LINEAR, 3-SLOTS, and 5-SLOTS in terms of the total number of slots offered and in terms of second-day slots. Interestingly, although COBB-DOUGLAS withholds feasible slots from the customer, if serving them incurs a high opportunity cost, in the long run, it offers many more slots. Finally, note that 3-SLOTS offers the lowest number of third-day slots, which is expected since these slots are characterized by low expected utilities for customers.

Figure 6 displays the average utility of the assortments for each demand rate and each tested policy. As in Figure 5, the presented measures are calculated based only on customers whose offered assortments are nonempty. In our view, this is the best measure of the quality of service, as perceived by customers who are not rejected.

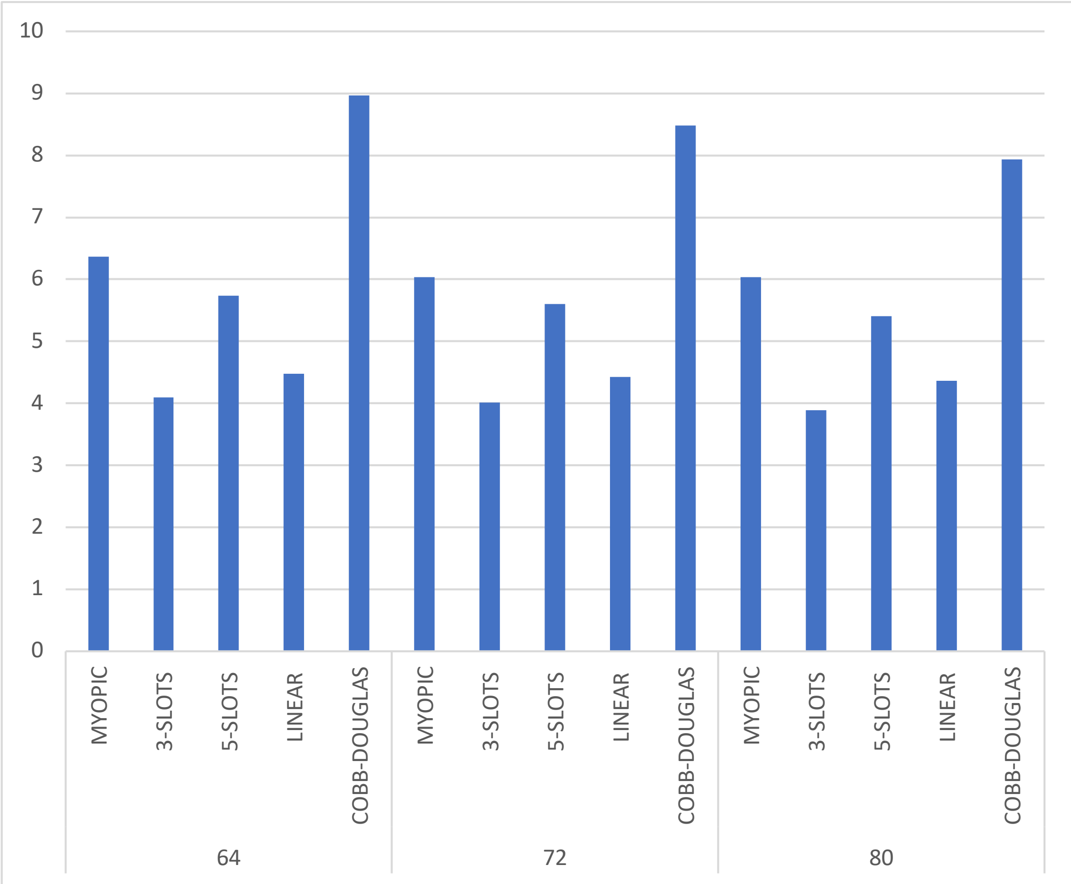


Figure 6: Utility of the offered assortments

COBB-DOUGLAS induces the highest utility. This phenomenon is expected because the number of offered slots, as well as the number of first-day and second-day slots, is the highest under this policy. MYOPIC is second best, by a large margin. Overall, the ranking between policies here seems to resemble that of Figure 5. Interestingly, COBB-DOUGLAS not only surpasses the other policies in terms of the acceptance rate but also results in offers that are on average much more attractive than those of the other policies, including LINEAR.

Figure 7 displays the average utility of the actual service slot for the five tested policies and three demand rates. The data are calculated based on the customers that are served.

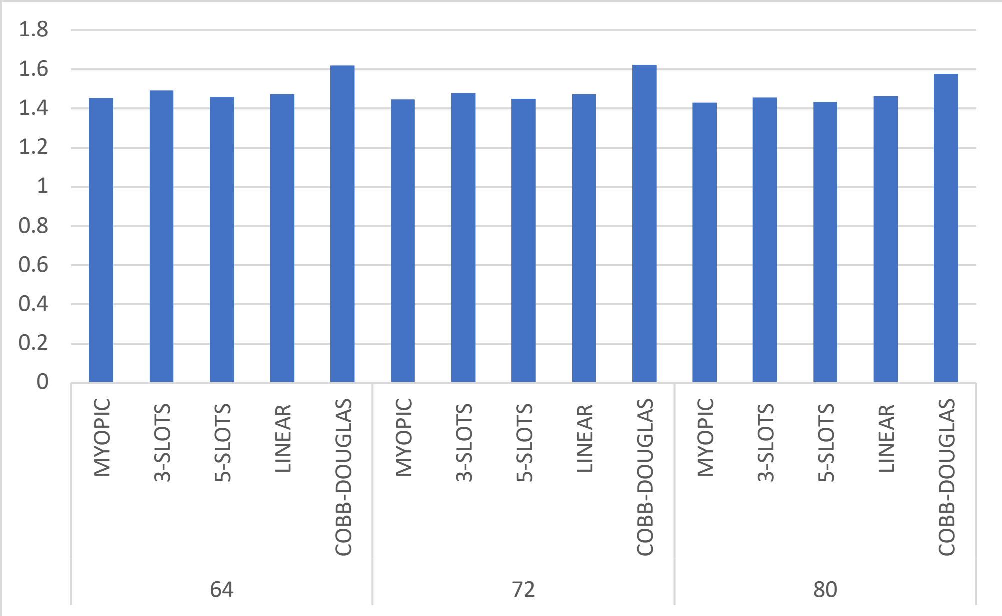


Figure 7: Average utility of actual service slot

COBB-DOUGLAS outperforms all other policies quite clearly for all demand rates. That is, not only does COBB-DOUGLAS succeed in maximizing the rate of served (accepted) customers, it also maximizes the actual service level for customers. Interestingly, all other policies, including LINEAR, produce a very similar average utility of service slots.

Figure 8 displays the proportion of customers served on the first and second days for the three demand rates and the five tested policies. Note that only the accepted customers in each policy are considered.

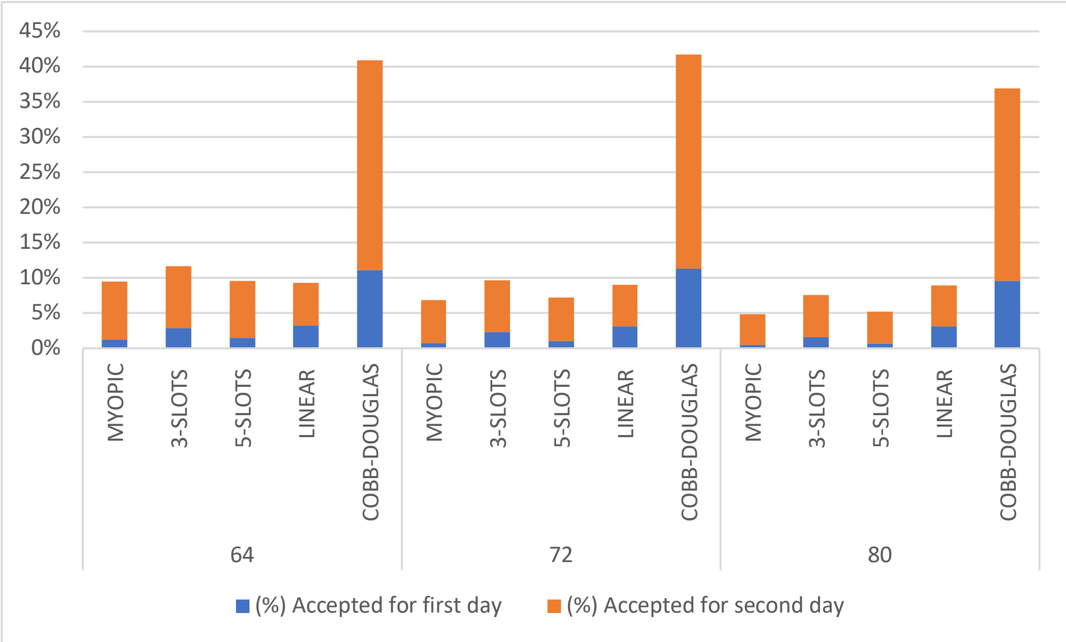


Figure 8: Customers accepting service on the first and second days by policy and demand rate

It is apparent from Figure 8 that the proportions of customers served on the first and second days are the highest for COBB-DOUGLAS, which further supports the superiority of this policy in terms of the perceived quality of service.

We conclude Section 5.2 by summarizing our main findings:

1. COBB-DOUGLAS leads to significantly better routes and schedules, as reflected by the larger number of accepted customers and average travel time of the vehicles per request. LINEAR, which also applies an advanced assortment optimization scheme, is the second best in terms of acceptance rate and comparable in terms of routing costs per customer, but mainly because it does not serve as many customers as COBB-DOUGLAS.
2. COBB-DOUGLAS generates the best service level among the tested policies according to all the measures that we calculated. When using COBB-DOUGLAS, more customers are accepted for service, and those who are accepted are served, on average, in more desirable time slots.

In other words, COBB-DOUGLAS induces bookings, schedules, and routes that are closer than all the other policies to the efficiency frontier between the number of served customers and the quality of service provided to the accepted customers.

5.3 The importance of the model components

In this section, we verify that all three arguments (RTS, RTR, and TT) of the opportunity cost function f are essential by applying our method while eliminating each of the arguments, one at a time. These reduced models can be viewed as special cases of the full model, with α, β, or γ fixed at 0. Figure 9 presents the estimated acceptance ratio for these three modified models for each of the 27 instances. The results of the full model are displayed as well. The performances of the three partial models are reported in detail in the electronic appendix.

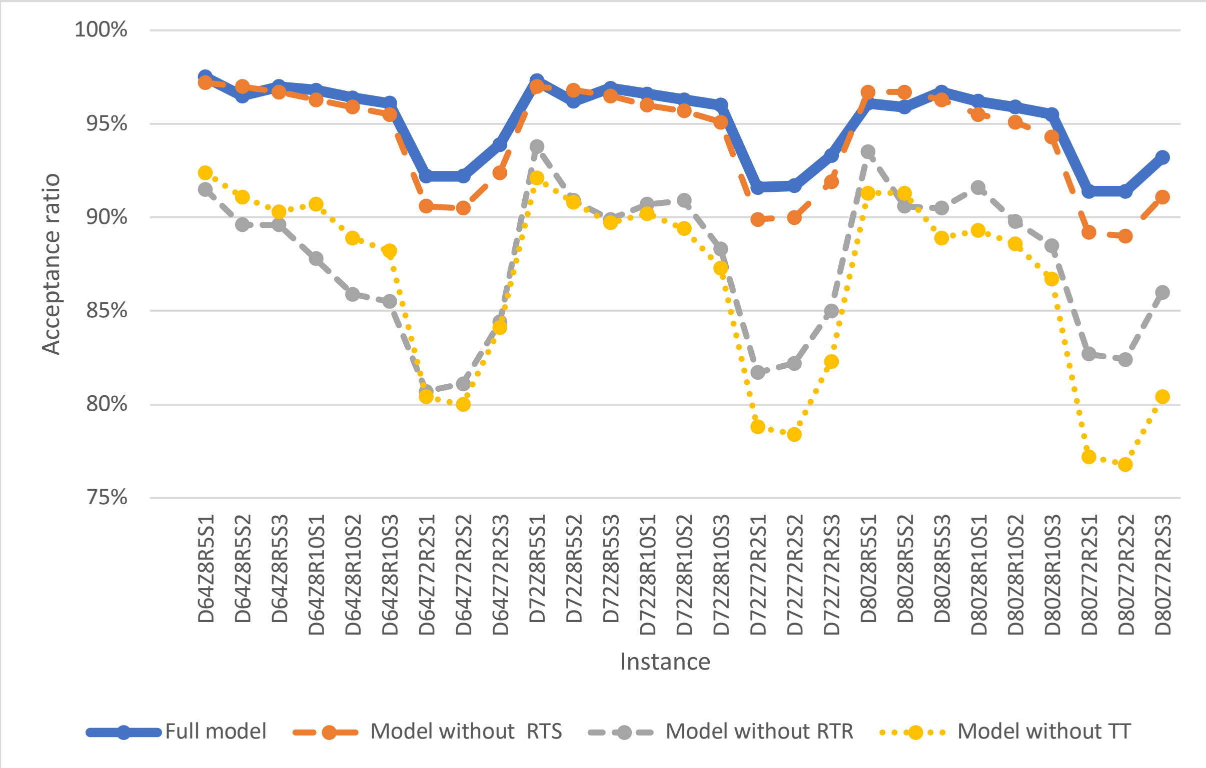


Figure 9: Acceptance ratio of the full model and its special cases

The full model obtains significantly better average results than each of the partial models (p-value < 0.001 in paired t-tests that compare the full model with each of the partial ones). It appears that the marginal effect of the RTS measure on the performance of the model is relatively moderate. In contrast, the effect of the TT and RTR measures is considerably more notable, especially in rural settings.

5.4 The merits of a multiday service horizon

Recall that the proposed booking mechanism allows technician appointment scheduling several days into the future with multiple cutoff times. In contrast, previous literature focused on a service horizon that consists of a single day and overlooked the interactions between consecutive working days. In this section, we examine the merits of considering a multiday service horizon. To this end, we repeat our main experiment for a subset of 9 out of 27 instances, namely, the instances for which $S = 1$, with two additional service horizon lengths: one day and five days.

Figure 10 displays the average proportion of abandoning and rejected customers for the five tested policies and the three lengths of the service horizon.

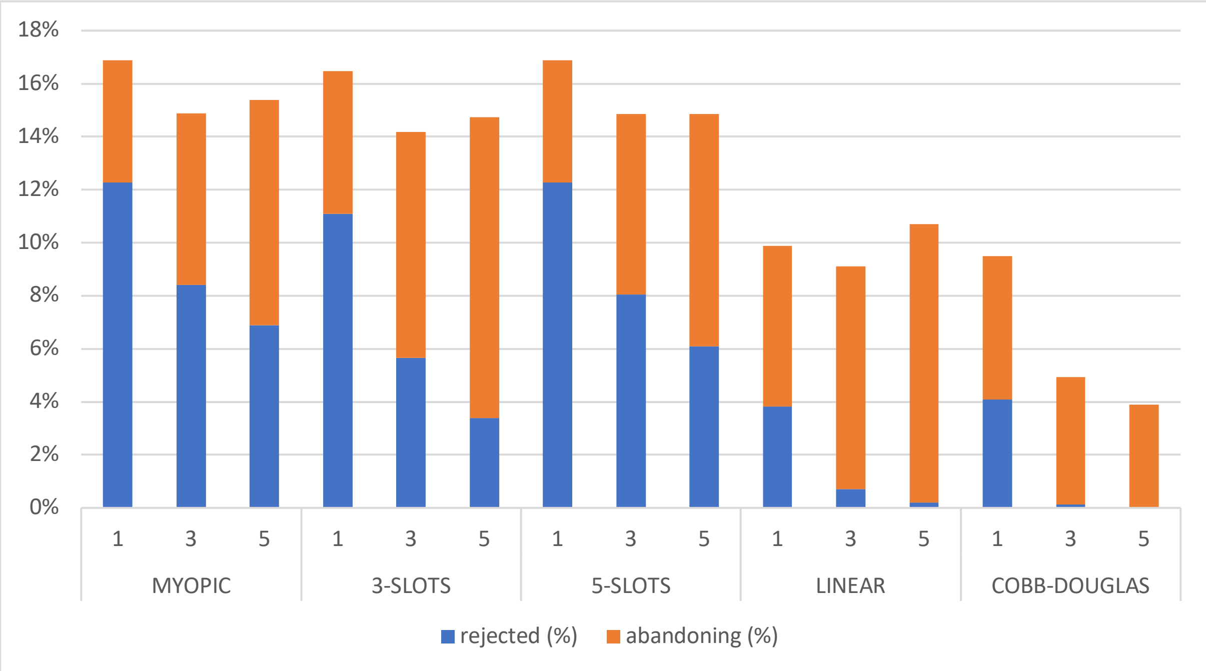


Figure 10: The proportion of rejected and abandoning customers for different service horizons

This experiment verifies that the advantage of COBB-DOUGLAS is maintained for different lengths of the service horizon. For all policies, increasing the service horizon strictly decreases the proportion of customers with an empty assortment, i.e., the rejected customers. This finding is expected since increasing the service horizon increases the number of feasible slots. However, the proportion of customers that abandon the process strictly increases for all policies other than COBB-DOUGLAS. Since the proportion of unserved customers is the sum of the abandoning and rejected customers, the best length of the service horizon for maximizing the acceptance rate is three days rather than five for all policies other than COBB-DOUGLAS. In other words, COBB-DOUGLAS is best suited to exploit longer service horizons.

The strict increase in the proportion of abandoning customers is caused by the structure of the assortment presented to them. Figure 11 displays the average cardinality of the assortments for the five policies and three lengths of the service horizon. As above, the figure is plotted based only on customers whose offered assortment is nonempty.

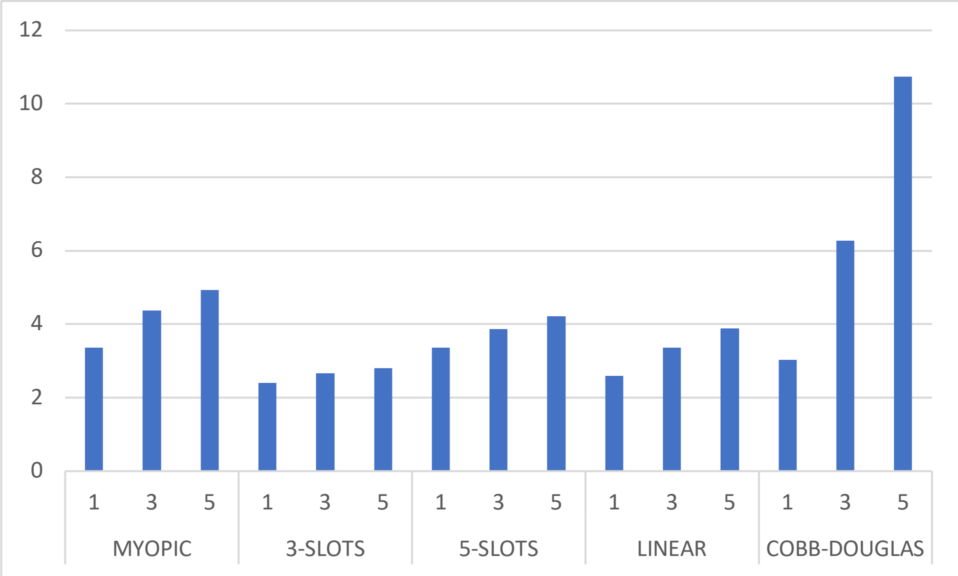


Figure 11: The size of an average assortment

Figure 11 demonstrates that increasing the service horizon has little effect on the size of the assortments for all policies other than COBB-DOUGLAS. For the 3-SLOTS policy, the effect is almost negligible, which follows from the fact that the number of offered slots is already limited to a very small value under this policy. The significant increase in the size of the assortments noted only for COBB-DOUGLAS demonstrates that it is the best policy for controlling the demand in a multiday setting.

Finally, Figure 12 displays the utility of the offered assortments for all policies and lengths of the service horizon. As usual, the figure is plotted based on customers who are not rejected.

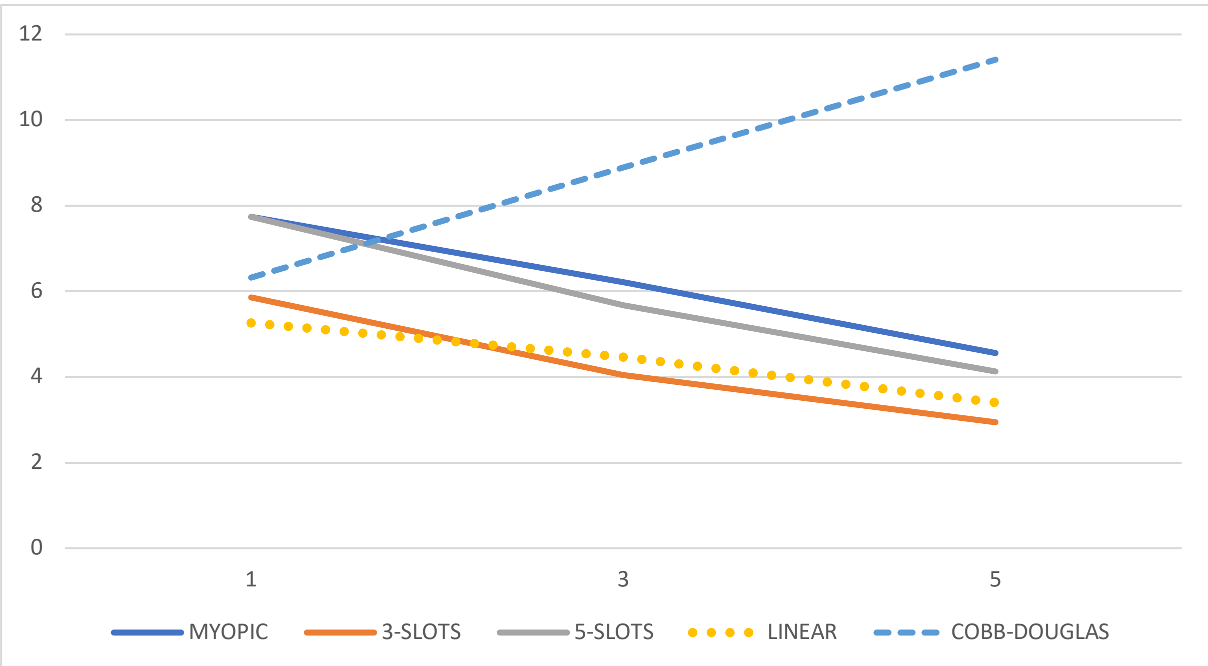


Figure 12: Utility of assortments for all policies and service horizons

In Figure 12, we observe that the effect of the service horizon length on the quality of the offered assortments depends on the policy. Greedier policies tend to "sell" all attractive slots early and, in the long run, can offer the poorest slots only. Hence, under these policies, the total utility decreases with the length of the service horizon. A similar but milder effect is observed in the LINEAR policy. Our proposed COBB-DOUGLAS is the only policy to exploit the additional flexibility obtained by a longer service horizon to improve the quality of the average offer made to customers. MYOPIC and 5-SLOTS generate offers with higher utility in a single-day setting, but recall that their acceptance rates are much lower.

We conclude Section 5.4 with a summary of our findings. We observed that COBB-DOUGLAS is the policy best suited to exploit the multiday setting. Under this policy, a longer service horizon allows a higher customer acceptance rate, while under the other tested policies, the effect of a longer service horizon is not consistent. The other policies create offers that are either nonattractive for customers or lead to the inefficient utilization of technicians, which in the long run results in poor offers.

6 Conclusions

In the past, the booking of field service personnel was usually performed through a sequential interaction between customers and service representatives in call centers. The current practice is increasingly based on using online services (such as web or mobile apps), which can conveniently present customers with all the offered alternatives at once. In this case, the process of determining the alternatives to offer requires an algorithmic approach.

In this study, we present an effective framework that addresses this challenge in realistic settings when the service horizon spans several days with different cutoff times and when the price of the service is exogenous and independent of the service time slot. Previous studies focused on the single cutoff case and overlooked the interactions between consecutive days. When the service provider can accept requests several days in advance, the decisions made on different days are interrelated. In this case, the problem is no longer separatable by working days. Therefore, the well-studied objective of maximizing the number of accepted requests (during a single planning period) should be replaced to maximize the steady-state acceptance rate.

The idea of the proposed method is to estimate the opportunity cost of accepting a request in each feasible slot based on a compact approximated representation of the state of the system. Using these costs, the provider optimizes the offered assortment of slots for each service request, assuming a discrete choice model representing the customers' preferences. The successful implementation of the proposed methods is highly dependent on a reliable forecast of future demand and a sufficiently accurate modeling of the user choice model.

A closely related line of research considers booking home-attended deliveries. In this case, the booking system should also consider the capacity constraint of the vehicles. Introducing this constraint into the proposed method requires including the remaining capacity in the state representation used for estimating the opportunity cost. It also straightforwardly affects the routing module.

The literature regarding the booking of field service personnel in multiday settings is still sparse, and we believe that the following extensions of our model require further investigation:

(1) Consider several classes of customers that differ in their time preferences, value to the service provider, required technician skills, and expected service duration. Each class may have a different optimal booking policy.

(2) If the provider has no capacity to serve a request during the service horizon, it may be extended to make additional service slots available. An interesting direction for future research is to introduce decisions regarding the length of the service horizon into the model and allowing its dynamic extension.

(3) Our solution approach makes a simplifying assumption that the choice model of the customers is fixed for each day of the week. In practice, the preferences of customers can be affected by seasonality and weekly cycles. For example, customers may prefer weekend time slots over weekdays. Introducing seasonality and day-of-the-week effects would require additional parameters for the function that estimates the opportunity cost. While this is possible using the method proposed in this study, much more computational effort will be required to fit the parameters correctly. A future study to meet this challenge will have to devise a much more efficient method to solve the parameter fitting problem.

(4) Concurrency issues should be resolved. When a large service provider performs the booking process, the system is likely to interact simultaneously with several customers. The slots offered to a particular customer may or may not be available to others while the interaction commences. The optimal booking policy may be affected by these concurrency considerations.

Acknowledgments

This research was supported by a grant from the Israel Science Foundation (ISF) [Grant no. 1367/17].

The first author is partially supported by a scholarship from the Shlomo-Shmeltzer institute for smart transportation in Tel Aviv University.

References

1. Niels Agatz, Ann Campbell, Moritz Fleischmann, Martin Savelsbergh (2011) Time Slot Management in Attended Home Delivery. Transportation Science 45(3):435-449.
2. Maria Albareda-Sambola, Elena Fernández, Gilbert Laporte (2014) The dynamic multiperiod vehicle routing problem with probabilistic information. Computers & Operations Research, 48, 31–39.
3. Enrico Angelelli, Nicola Bianchessi, Renata Mansini, M. Grazia Speranza (2009) Short term strategies for a dynamic multi-period routing problem. Transportation Research Part C: Emerging Technologies, 17(2): 106–119.
4. Kursad Asdemir, Varghese S. Jacob, Ramayya Krishnan (2009) Dynamic pricing of multiple home delivery options. European Journal of Operational Research 196: 246-257.
5. Edison Avraham, Tal Raviv (2020) The data-driven time-dependent traveling salesperson problem. Transportation Research Part B: Methodological 134: 25-40.
6. Russell W. Bent, Pascal Van Hentenryck (2004) Scenario-Based Planning for Partially Dynamic Vehicle Routing with Stochastic Customers. Operations Research 52(6):977-987.
7. Kris Braekers, Katrien Ramaekers, Inneke Van Nieuwenhuyse (2016) The vehicle routing problem: State of the art classification and review. Computers & Industrial Engineering 99: 300-313.

8. Dominic Bühler, Robert Klein, Michael Neugebauer (2016) Model-based delivery cost approximation in attended home services. Computers & Industrial Engineering 98: 78-90.
9. Ann Melissa Campbell, Martin W. P. Savelsbergh (2005) Decision Support for Consumer Direct Grocery Initiatives. Transportation Science 39(3):313-327.
10. Ann Melissa Campbell, Martin Savelsbergh (2006) Incentive Schemes for Attended Home Delivery Services. Transportation Science 40(3):327-341.
11. Xi Chen, Barrett W. Thomas, Mike Hewitt (2017) Multi-period technician scheduling with experience-based service times and stochastic customers. Computers & Operations Research, 82, 1–14.
12. Catherine Cleophas, Jan Fabian Ehmke (2014) When Are Deliveries Profitable? Business & Information Systems Engineering 6(3):153-163.
13. Jan Fabian Ehmke, Ann Melissa Campbell (2014) Customer acceptance mechanisms for home deliveries in metropolitan areas. European Journal of Operational Research 233: 193-207.
14. Guillermo Gallego, Richard Ratliff, Sergey Shebalov (2015) A General Attraction Model and Sales-Based Linear Program for Network Revenue Management Under Customer Choice. Operations Research 63(1):212-232.
15. Gianpaolo Ghiani, Emanuele Manni, Antonella Quaranta, Chefi Triki (2009) Anticipatory algorithms for same-day courier dispatching. Transportation Research Part E: Logistics and Transportation Review 45(1): 96-106.
16. Gianpaolo Ghiani, Emanuele Manni, Barrett W. Thomas (2012) A Comparison of Anticipatory Algorithms for the Dynamic and Stochastic Traveling Salesman Problem. Transportation Science 46(3):374-387.
17. Florent Hernandez, Michel Gendreau, Jean-Yves Potvin (2017) Heuristics for tactical time slot management: a periodic vehicle routing problem view. International Transactions In Operational Research. 24(6):1233–1252.
18. Robert Klein, Sebastian Koch, Claudius Steinhardt, Arne K. Strauss (2020) A review of revenue management: Recent generalizations and advances in industry applications. European Journal of Operational Research 284: 397-412.
19. Robert Klein, Jochen Mackert, Michael Neugebauer, Claudius Steinhardt (2018) A model-based approximation of opportunity cost for dynamic pricing in attended home delivery. OR Spectrum 40(4): 969-996.
20. Robert Klein, Michael Neugebauer, Dimitri Ratkovitch, Claudius Steinhardt (2019) Differentiated Time Slot Pricing Under Routing Considerations in Attended Home Delivery. Transportation Science 53(1):236-255.
21. Sebastian Koch, Robert Klein (2020) Route-based approximate dynamic programming for dynamic pricing in attended home delivery. European Journal of Operational Research 287: 633-652.
22. Charlotte Köhler, Jan Fabian Ehmke, Ann Melissa Campbell (2020) Flexible time window management for attended home deliveries. Omega 91:102023
23. Jochen Mackert (2019) Choice-based dynamic time slot management in attended home delivery. Computers & Industrial Engineering 129: 333-345.

24. Arturo E. Pérez Rivera, Martijn R.K. Mes (2017) Anticipatory freight selection in inter- modal long-haul round-trips. Transportation Research Part E: Logistics and Transportation Review, 105, 176–194.
25. Victor Pillac, Michel Gendreau, Christelle Guéret, Andrés L. Medaglia (2013) A review of dynamic vehicle routing problems. European Journal of Operational Research 225 (1): 1-11.
26. Warren B. Powell (2019) A unified framework for stochastic optimization. European Journal of Operational Research 275 (3): 795-821.
27. Harilaos N. Psaraftis, Min Wen, Christos A. Kontovas (2016) Dynamic vehicle routing problems: Three decades and counting. Networks **67**(1) 3-31.
28. María I. Restrepo, Frédéric Semet, Thomas Pocreau (2019) Integrated Shift Scheduling and Load Assignment Optimization for Attended Home Delivery. Transportation Science 53(4):1150-1174.
29. Sheldon Ross (2013), Simulation (Fifth Edition), Academic Press.
30. Ninja Soeffker, Marlin Ulmer, Dirk Mattfeld (2017) On Fairness Aspects of Customer Acceptance Mechanisms in Dynamic Vehicle Routing. Proceedings of Logistikmanagement 2017: 17-24.
31. Marius M. Solomon (1987) Algorithms for the vehicle routing and scheduling problem with time window constraints. Operations Research 35(2): 54-65.
32. Arne Strauss, Nalan Gülpınar, Yijun Zheng (2020) Dynamic pricing of flexible time slots for attended home delivery. European Journal of Operational Research https://doi.org/10.1016/j.ejor.2020.03.007.
33. Arne K. Strauss, Robert Klein, Claudius Steinhardt (2018) A review of choice-based revenue management: Theory and methods. European Journal of Operational Research 271: 375-387.
34. Marlin W. Ulmer, Dirk C. Mattfeld, Felix Köster (2018a) Budgeting Time for Dynamic Vehicle Routing with Stochastic Customer Requests. Transportation Science 52(1):20-37.
35. Marlin W. Ulmer, Ninja Soeffker, Dirk C. Mattfeld (2018b) Value function approximation for dynamic multi-period vehicle routing. European Journal of Operational Research 269: 883-899.
36. Marlin W. Ulmer (2019) Anticipation versus reactive reoptimization for dynamic vehicle routing with stochastic requests. *Networks* 73:277–291.
37. Marlin W. Ulmer, Barrett W. Thomas (2019) Meso-parametric value function approximation for dynamic customer acceptances in delivery routing. European Journal of Operational Research 285(1): 183-195.
38. Marlin W. Ulmer (2020) Dynamic Pricing and Routing for Same-Day Delivery. Transportation Science 54(4):1016-1033
39. Stacy A. Voccia, Ann Melissa Campbell, Barrett W. Thomas (2019) The Same-Day Delivery Problem for Online Purchases. Transportation Science 53(1):167-184.
40. Min Wen, Jean-François Cordeau, Gilbert Laporte, Jesper Larsen (2010) The dynamic multi-period vehicle routing problem. Computers & Operations Research, 37(9): 1615–1623.
41. Xinan Yang, Arne K. Strauss, Christine S. M. Currie, Richard Eglese (2016) Choice-Based Demand Management and Vehicle Routing in E-Fulfillment. Transportation Science 50(2):473-488.
42. Xinan Yang, Arne K. Strauss (2017) An approximate dynamic programming approach to attended home delivery management. European Journal of Operational Research 263: 935-945.