Approximate Dynamic Programming for a Dynamic Appointment Scheduling Problem

Zlatana Nenova
Daniels College of Business, University of Denver
zlatana.nenova@du.edu

Manuel Laguna • Dan Zhang
Leeds School of Business, University of Colorado Boulder
{laguna, dan.zhang}@colorado.edu

We study a dynamic appointment scheduling problem with cancellations and overbooking motivated by a medical clinic, where appointment requests arrive over time. The objective is to balance patient waiting time with service providers' overtime and idle time. The problem is formulated as a finite-horizon stochastic dynamic program. However, the formulation suffers from the curse of dimensionality as the state is the service schedule, which is inherently high dimensional. We propose a solution approach based on approximate policy iteration and value function approximation. We validate the approach with data from a public hospital in the US. We use the data to develop a Weibull accelerated failure time model to estimate the time- and patient-dependent cancellation and no-show probabilities. Our solution approach allows treating each patient as his or her own class. The approximate policy iteration approach is simulation-based and can accommodate complex system dynamics. Our numerical study shows that the approach is competitive against several computational benchmarks.

This version: March 10, 2020

1. Introduction

In medical services, it is common to book appointments weeks, or even months, in advance. Due to the long appointment lead time, a scheduled appointment may be cancelled ahead of time. In addition, patients may fail to show up (no-show) for their appointments. If not properly managed, appointment cancellations and no-shows can lead to the under-utilization of doctors, as they will have schedules with empty time slots. To compensate for cancellations and no-shows, many hospitals overbook appointments. However, in the event of unexpectedly high patient show-ups, overbooking can lead to excessive patients' waiting time and doctors' overtime. Service providers want to make appointment decisions that account for rejection costs (e.g. cost of goodwill), patients' waiting costs, doctors' idle time, and overtime costs.

We consider a dynamic appointment scheduling problem where appointment requests arrive over time, and patients with reservations may cancel or no-show for their appointments. Patients can differ in their rejection and waiting costs, as well as cancellation and no-show probabilities. We

formulate the problem as a finite-horizon stochastic dynamic program. A major challenge in solving the stochastic dynamic program is the well-known "curse of dimensionality" caused by the state space growing exponentially in the number of appointment slots and patient classes. Unlike other applications, such as the airline revenue management with overbooking and cancellations (Subramanian et al. 1999), the cost evaluation of a fixed appointment schedule is non-trivial. Furthermore, scheduling appointments is complex even when patients are homogeneous because of the intricate system dynamics (Robinson and Chen 2010). Due to the complexities associated with solving the problem exactly, the literature focuses on approximation methods, broadly classified as approximate dynamic programming (ADP).

We propose an approximate policy iteration (API) algorithm (Bertsekas and Tsitsiklis 1996, Powell 2007, Bertsekas 2011), which can accommodate the nonlinear structure of the terminal value function. API is a broad class of ADP methods that use iterative simulations to learn the dynamic programming value functions. We validate the API algorithm on data obtained from a large public hospital in the US. Rather than clustering patients prior to scheduling, we develop and evaluate a personalized scheduling model, where each patient is treated as his or her own class. We estimate the arrival patterns of appointment requests, cancellation probabilities of existing appointments, and no-show probabilities for each patient. In particular, we use a Weibull accelerated failure time model (Klein and Moeschberger 2006) to estimate the cancellation and no-show probabilities. The model is parametric and dynamic, and the estimated cancellation and no-show probabilities are time- and patient-dependent.

We perform a simulation study to evaluate the proposed API algorithm. Since it is not possible to compare the performance of the algorithm against an optimal policy, we propose a lower bound on the expected total cost based on ideas from information relaxation, where the cost is evaluated by assuming that arrivals and cancellations are observed but no-shows are not. The lower bound provides a benchmark to evaluate the optimality gap of the policy based on the proposed API algorithm. We also consider several scheduling policies, which provide upper bounds on the cost. Finally, we compare our model to the heuristic policy presented in Section 6 of Zacharias and Pinedo (2014).

We contribute to the appointment scheduling literature by generalizing the expected terminal cost evaluation in Robinson and Chen (2010) and Zacharias and Pinedo (2014) for multiple patient classes. Our model can be viewed as a generalization of the single-leg revenue management problem with overbooking and cancellation in Subramanian et al. (1999). We demonstrate that our model outperforms several alternative policies. In particular, it outperforms the policy proposed by Zacharias and Pinedo (2014) in 77% of the tested settings.

The remainder of this paper is organized as follows. We review the literature in Section 2. Section 3 presents the dynamic programming formulation. We introduce our API algorithm in Section 4. Section 5 presents several benchmark policies. Section 6 discusses parameter estimation. Section 7 presents our numerical results and their implications. Section 8 concludes.

2. Literature Review

In this section, we review the relevant literature with a focus on dynamic appointment scheduling and approximate dynamic programming.

In recent years, the operations research community has paid significant attention to scheduling problems in the medical industry (Cayirli and Veral 2003, Mondschein and Weintraub 2003, Gupta and Denton 2008, Ahmadi-Javid et al. 2017). Many papers in the appointment scheduling literature consider *static* appointment scheduling where the system dynamics is simplified or completely ignored (Kaandorp and Koole 2007, Hassin and Mendel 2008, Klassen and Yoogalingam 2009, Robinson and Chen 2010). More recent work in this area studies *dynamic* appointment scheduling problems where the system dynamics is explicitly modeled and analyzed (Liu et al. 2010, Zacharias and Pinedo 2014). Some researchers consider systems with enriched system dynamics, such as patients' appointment time preferences (Gupta and Wang 2008, Wang and Gupta 2011, Feldman et al. 2012).

Dynamic programming has been widely used in healthcare operations, see Alagoz et al. (2010) for a review and Kolesar (1970) for an early reference. Grant et al. (2017) study the trade-off between delaying an appointment at the risk of costly failures versus the additional cost of scheduling the appointment sooner using surge capacity. They show that some simple policies can be optimal under certain technical assumptions. When these assumptions are violated, they evaluate the dynamic programming formulation exactly using a numerical procedure. Other recent papers that use dynamic programming models for appointment scheduling problems include LaGanga and Lawrence (2007), Patrick et al. (2008), Liu et al. (2010), Gocgun et al. (2011), LaGanga and Lawrence (2012), Patrick (2012), Sauré et al. (2012), Schütz and Kolisch (2012), Barz and Rajaram (2013), Samorani and LaGanga (2013), Gocgun and Puterman (2014), and Samiedaluie et al. (2014).

As noted in Section 1, we use an ADP method to tackle the "curse of dimensionality" when scheduling appointments (Bertsekas and Tsitsiklis 1996, Powell 2007, Bertsekas 2011). Most of the aforementioned articles use linear programming-based ADP (LP-based ADP) (Adelman 2003, 2004, de Farias and Van Roy 2004, de Farias and Van Roy 2006). The linear programming formulation for dynamic programs originated in the 1960s; see Puterman (1994) for a discussion of the early literature. The LP-based ADP starts with the linear programming formulation of the corresponding

dynamic program and typically assumes specific parametric forms for the value functions (Schweitzer and Seidmann 1985). The parameterization of the value function helps reduce the number of decision variables because only function coefficients need to be determined instead of the full set of function values. Nevertheless, the resulting approximate linear programs (ALPs) are still quite challenging to solve because they are typically accompanied by a significant number of constraints. The focus of this literature has been on applying specialized algorithms, such as column generation or constraint sampling, to solve the ALPs. For the dynamic appointment scheduling problem considered in our paper, it is not clear whether the LP-based ADP can be immediately applicable as the terminal value function is highly nonlinear.

API algorithms possess solid theoretical properties when a lookup table representation of the value function is used. Even though the theoretical development of APIs is incomplete when the value function approximations are used, there is ample evidence of strong performance in practice. Sauré et al. (2015) propose an API algorithm based on a post-decision state formulation and a logistic value function approximation, which is applied to a multi-priority patient scheduling problem. Their focus is to compare the solution quality and policies from the API algorithm with the LP-based ADP (Patrick et al. 2008). They show that the policies from the API algorithm have several advantages over the policies produced by the LP-based ADP. More recently, Koch (2017) shows that API combined with some well-known value function approximations leads to policies that are competitive when compared to solution methods based on LP-based ADP for the network revenue management problem. As LP-based ADP cannot be directly applied in our setting, we build computational benchmarks based on the heuristic from Zacharias and Pinedo (2014) and show that the API algorithm can obtain higher quality solutions. Liu et al. (2010) propose an API algorithm in a similar setting. Unlike us, they only evaluate their API algorithm once. Our numerical study shows that performing multiple iterations is crucial for obtaining high quality solutions.

3. Model Formulation

This section introduces a finite-horizon dynamic programming model for a dynamic appointment scheduling problem. We consider a medical clinic with N appointment slots per day. We assume that appointments are made in advance. The booking horizon is divided into T discrete time periods. Time counts forward, so that the first period is period 1 and the last period is period T. Medical appointments occur in period $T+1$. Patients are classified into K classes, which have different arrival rates, cancellation probabilities, and rejection costs. We assume that there is at most one event occurring in the booking period t, which can be either the arrival of an appointment request or an appointment cancellation. The arrival rate of a patient from class k in period t is λ_{tk}. An existing

appointment in class k will be canceled in period t with probability q_{tk}. A rejected appointment request from class k in period t incurs a cost r_{tk}. In period $T+1$, each class-k patient shows up with a probability of $1-q_{T+1,k}$.

The finite-horizon stochastic dynamic programming model can be stated as follows. The state is an $N \times K$ matrix $\mathbf{X} = (x_{nk})$, where x_{nk} is the number of class k patients scheduled in appointment slot n. Without loss of generality, we assume that the number of patients booked in slot n is bounded by a number b_n. We require that

$$\sum_k \lambda_{tk} + \sum_n b_n \max_k q_{tk} \leq 1, \qquad \forall t.$$

Let $V_t(\mathbf{X})$ denote the total expected cost given state $\mathbf{X}$ in period t. For each t and $\mathbf{X}$, the dynamic programming equation is given by

$$\begin{aligned} V_t(\mathbf{X}) &= \sum_k \lambda_{tk} \min\left\{ r_{tk} + V_{t+1}(\mathbf{X}), \min_n V_{t+1}(\mathbf{X}+e_{nk}) \right\} + \sum_k \sum_n q_{tk} x_{nk} V_{t+1}(\mathbf{X}-e_{nk}) \\ &\quad + \left(1 - \sum_k \lambda_{tk} - \sum_k \sum_n q_{tk} x_{nk}\right) V_{t+1}(\mathbf{X}) \\ &= \sum_k \lambda_{tk} \min\left\{ r_{tk}, \min_n \Delta_{nk} V_{t+1}(\mathbf{X}) \right\} - \sum_k \sum_n q_{tk} x_{nk} \Delta_{nk} V_{t+1}(\mathbf{X}-e_{nk}) + V_{t+1}(\mathbf{X}). \end{aligned} \tag{1}$$

In the above, e_{nk} is a matrix where the (n,k)-th entry is one and all other entries are zeros and

$$\Delta_{nk} V_{t+1}(\mathbf{X}) = V_{t+1}(\mathbf{X}+e_{nk}) - V_{t+1}(\mathbf{X}), \qquad \forall n, k, \mathbf{X}.$$

We also define the following notations:

- α: unit waiting cost for all patients.
- β: the overtime premium for doctors; $\beta = 50\%$ by default.
- D: the doctor's expected length of day.
- I: the doctor's expected idle time before the end of day.
- O: the doctor's expected overtime.
- W: the expected waiting time of the patients.

Thus, the total cost of a schedule $\mathbf{X}$ is given by

$$V_{T+1}(\mathbf{X}) = I + \alpha W + \beta O. \tag{2}$$

We discuss how to evaluate the terminal value (2) in the next section.

Before proceeding, we point out that the optimal policy is relatively simple. We state the result without a proof in the following proposition.

PROPOSITION 1. *Given state* $\mathbf{X}$ *in period* t*, it is optimal to schedule a class-*k *patient in slot* n^* *if*

$$r_{tk} \leq \min_n \Delta_{nk} V_{t+1}(\mathbf{X}), \quad n^* \in \arg\min_n \Delta_{nk} V_{t+1}(\mathbf{X}).$$

Otherwise, it is optimal to reject the patient.

The structure of the optimal policy in Proposition 1 is implied by the optimality equation 1. Proposition 1 states that it is optimal to schedule a patient to a slot with the smallest opportunity cost, which should be lower than the rejection cost. Otherwise, it is optimal to reject the patient. Therefore, once the value function $V_t(\mathbf{X})$ is determined, it is trivial to compute the optimal policy.

4. An Approximate Policy Iteration (API) Algorithm

We introduce an API algorithm with value function approximation to solve the problem (1). The main challenge in solving the problem stems from the high dimensionality of the state variable $\mathbf{X}$, which is an $N \times K$ matrix. In order to address the state space explosion, we introduce a value function approximation based on least squares regression. The heart of policy iteration is the so-called policy improvement step. In order to reduce the number of policy-improvement iterations, it is important to start with a reasonably strong initial policy. We describe our procedure in the following sections. Section 4.1 briefly describes the procedure used to generate sample paths. We discuss the initialization in Section 4.2. Section 4.3 introduces the value function approximation based on least-squares regression. Section 4.4 describes the API algorithm in detail. We emphasize that the API algorithm is simulation-based and therefore can incorporate rather complex system dynamics. Furthermore, it is decoupled from the value function approximation architecture and can be used with different value function approximations.

4.1 Generating Sample Paths

A sample path can be denoted by the vector $\mathbf{u} = (u_1, \ldots, u_T)$, where u_t is a random number between 0 and 1 for each t. The value of u_t can be used to determine whether there is an arrival or a cancellation from a class k patient. If $u_t \in [0, \lambda_{t1})$, then there is a class-1 customer arrival. For $k = 2, \ldots, K$, if $u_t \in [\sum_{j=1}^{k-1} \lambda_{tj}, \sum_{j=1}^{k} \lambda_{tj})$, then there is a class-$k$ customer arrival. If $u_t \in [\sum_{j=1}^{K} \lambda_{tj}, \sum_{j=1}^{K} \lambda_{tj} + \sum_n q_{t1} x_{n1})$, then there is a class-1 cancellation. For $k = 2, \ldots, K$, if $u_t \in [\sum_{j=1}^{K} \lambda_{tj} + \sum_n \sum_{j=1}^{k-1} q_{tj} x_{nj}, \sum_{j=1}^{K} \lambda_{tj} + \sum_n \sum_{j=1}^{k} q_{tj} x_{nj})$, then there is a class-$k$ cancellation. Since all class-k ($k = 1, \ldots, K$) customers have equal probability of cancellation, we can also use u_t to determine which customer from class k canceled his or her appointment.

4.2 Initialization

To ensure that the algorithm produces a high-quality scheduling policies within a small number of iterations, we need to start with a reasonably good initial policy. We use an initialization policy where all appointment requests are accepted. We also assume that all cancellations are observed prior to period $T+1$. Once all patients attending the clinic in period $T+1$ are known, we schedule them following the procedure in Proposition 1 of Zacharias and Pinedo (2014). The produced schedule is $\mathbf{X}$. Based on $\mathbf{X}$, we can calculate the expected cost $\tilde{V}_{T+1}(\mathbf{X})$ and the predictors presented in Section 4.3. We estimate $\tilde{V}_{T+1}(\theta_t, \mathbf{X})$ using a stepwise least squares regression model; see Algorithm 1. A reasonable initial policy can be computed by setting

$$\tilde{V}_t(\mathbf{X}) \approx \tilde{V}_{T+1}(\mathbf{X}), \qquad \forall t, \mathbf{X}.$$

In period $T+1$, each class-k patient shows up with a probability of $1-q_{T+1,k}$. We can generalize the model in Robinson and Chen (2010) and Zacharias and Pinedo (2014) to calculate the expected cost $V_{T+1}(\mathbf{X})$.

Before proceeding, we introduce some additional notations.

- $\mathbf{x}_n$: the n-th row of $\mathbf{X}$.
- $\mathbf{x}^k$: the k-th column of $\mathbf{X}$.
- $\mathbf{y}_n$: the vector of show-up appointments in slot n.
- $n_{\max} = \max\{n : \sum_k x_{nk} \geq 1\}$: the last time slot with at least one patient scheduled.
- $b(i,p,j)$: the probability that a binomial random variable with parameters i and p takes a value equal to j.
- $\mathbf{z}_n = (z_{n,1}, \ldots, z_{n,K})$: the vector denoting the number of patients in each class at the beginning of slot n (i.e. at the end of time slot $n-1$). We have $\mathbf{z}_1 = 0$.
- $k(\mathbf{z}_n) = \max\{k : z_{n,k} \geq 1\}$: the largest index of the positive element of a vector $\mathbf{z}_n$.
- $\mathbf{e}_k$: a vector with the k-th element 1 and all other elements 0.

The vector $\mathbf{z}_n$ satisfies

$$\begin{aligned} \mathbf{z}_1 &= \mathbf{0}, \\ \mathbf{z}_n &= \begin{cases} \mathbf{z}_{n-1} + \mathbf{y}_{n-1} - \mathbf{e}_{k(\mathbf{z}_{n-1}+\mathbf{y}_{n-1})}, & \text{if } \mathbf{z}_{n-1} + \mathbf{y}_{n-1} \neq \mathbf{0}, \\ \mathbf{0}, & \text{otherwise}, \end{cases} \qquad \forall n = 2, \ldots, N, \\ \mathbf{z}_{N+1} &= \mathbf{z}_N - \mathbf{e}_{k(\mathbf{z}_N)}. \end{aligned}$$

The probability that $\mathbf{y}_n$ out of $\mathbf{x}_n$ scheduled patients show up in period n equals:

$$P(\mathbf{y}_n) = \prod_k P(y_{n,k}) = \prod_k b(x_{n,k}, 1-q_{T+1,k}, y_{n,k}).$$

Let $P(\mathbf{z}_n = \mathbf{z}|\mathbf{z}_{n-1})$ be the probability that $\mathbf{z}$ individuals are in the office (system) at the beginning of slot n given that there are $\mathbf{z}_{n-1}$ patients in the system at the beginning of slot $n-1$. The probability that the office is not empty ($\mathbf{z} \neq \mathbf{0}$) at time n conditional on $\mathbf{z}_{n-1}$ can be represented as

$$P(\mathbf{z}_n = \mathbf{z}|\mathbf{z}_{n-1}) = P(\mathbf{z}_{n-1} + \mathbf{y}_{n-1} - \mathbf{e}_{k(\mathbf{z}_{n-1}+\mathbf{y}_{n-1})} = \mathbf{z}|\mathbf{z}_{n-1}) = P(\mathbf{y}_{n-1} - \mathbf{e}_{k(\mathbf{z}_{n-1}+\mathbf{y}_{n-1})} = \mathbf{z} - \mathbf{z}_{n-1}|\mathbf{z}_{n-1}).$$

Therefore, the difficulty of an explicit formula for the probability of $\mathbf{z}_n$ relates to the $\mathbf{e}_{k(\mathbf{z}_{n-1}+\mathbf{y}_{n-1})}$ term. However, this probability can be easily evaluated using simulation.

We assume that the doctor will leave after examining his or her last patient for the day. To determine whether there is at least one patient showing up in period $n_{\max}$, the doctor must stay in the office until the end of slot $n_{\max} - 1$. Furthermore, the expected length of a doctor's day will depend on the expected number of patients (i) not completed by the end of period $n_{\max-1}$ and (ii) arriving in time slot $n_{\max}$ given schedule $\mathbf{X}$.

Let $s(\cdot)$ denote the component sum of a vector, for example $s(\mathbf{z}_n) = \sum_{k=1}^{K} z_{n,k}$. We have

$$D = n_{\max} - 1 + \sum_{j=1}^{s(\mathbf{z}_{n_{\max}})} j \sum_{s(\mathbf{z})=j} P(\mathbf{z}_{n_{\max}} = \mathbf{z}) + \sum_{j=1}^{s(\mathbf{x}_{n_{\max}})} j \sum_{s(\mathbf{y})=j} P(\mathbf{y}_{n_{\max}} = \mathbf{y}).$$

The expected idle time equals the difference between a doctor's length of day (D) and his/her workload:

$$I = D - \sum_{k} (1 - q_{T+1,k}) \left(\sum_{n} x_{n,k} \right)$$

The expected overtime is calculated as the difference between the appointments' completion time (D) and overall length (N). Thus, it depends on N , $n_{\max}$, and the number of patients at the end of $n_{\max}$ (i.e. $\mathbf{z}_{n_{\max}+1}$).

$$O = \begin{cases} 0, & \text{if } 0 < n_{\max} \leq N - s(\mathbf{z}_{n_{\max}+1}), \\ \sum_{j=1}^{s(\mathbf{z}_{n_{\max}+1})-(N-n_{\max})} j \sum_{\langle \mathbf{i},\mathbf{z} \rangle = j} P(\mathbf{z}_{n_{\max}+1} = \mathbf{z}), & \text{if } N - s(\mathbf{z}_{n_{\max}+1}) < n_{\max} \leq N. \end{cases}$$

In the first case the overtime is 0 because the number of leftover appointments at the end of period $n_{\max}$ (i.e., $s(\mathbf{z}_{n_{\max}+1})$) is at most equal to the leftover slots between periods N and $n_{\max}$, which ensures that everybody will be seen by the end of the day. The second case corresponds to the number of leftover patients at the end of period $n_{\max}$ exceeding the number of leftover appointment slots, i.e. $s(\mathbf{z}_{n_{\max}+1}) > N - n_{\max}$. The equation quantifies the expected number of excess appointments (j). Because overtime cost does not depend on patient's class, we aggregate the probabilities across all $\mathbf{z}$ which contain the same total number of patients (i.e. $s(\mathbf{z}_{n_{\max}+1}) = j$).

Waiting time should be examined separately for patients at the end of slot $n_{\max}$ ($\mathbf{z}_{n_{max}+1}$) and at the end of slots 1 through $n_{\max} - 1$ ($\mathbf{z}_2, ..., \mathbf{z}_{n_{\max}}$). Without loss of generality, we can ignore the

class composition of patients waiting at the end of slot n, as the unit waiting cost (α) is the same for all k. The total expected patient waiting time is given by

$$W = \sum_{n=2}^{n_{\max}} \sum_{j=1}^{s(\mathbf{z}_n)} j \sum_{s(\mathbf{z}_n)=j} P(\mathbf{z}_n = \mathbf{z}) + \sum_{j=1}^{s(\mathbf{z}_{n_{\max}+1})} \sum_{l=1}^{j} l \sum_{s(\mathbf{z})=j} P(\mathbf{z}_{n_{\max}+1} = \mathbf{z}).$$

The total cost of a schedule $\mathbf{X}$ is given by $V_{T+1}(\mathbf{X})$ defined in (2).

Algorithm 1 API algorithm

for $s = 1$ to S **do**
  $\mathbf{X}_1^{(s)} \leftarrow \mathbf{0}$
  for $t = 1$ to T **do**
    $U_t \leftarrow$ Uniform$[0,1]$
    $u_t^* \leftarrow f(u_t, \boldsymbol{\lambda_t}, \mathbf{q_t})$ (see Section 4.1)
  end for
  Update $\mathbf{X}_T^{(s)}$ based on $u_t^*, t \in \{1,T\}$, $\mathbf{X}_1^{(s)}$ & following Proposition 1 in Zacharias and Pinedo (2014)
  if $\mathbf{X}_{T+1}^{(s)} = \mathbf{0}$ **then**
    $\tilde{V}_{T+1}^{(s)}(\mathbf{X}_{T+1}^{(s)}) \leftarrow 0$
  else
    Simulate no-shows based on $\mathbf{X}_{T+1}^{(s)}$ and $\mathbf{q}_{T+1}$
    Serve the patients that show-up in $\mathbf{X}_{T+1}^{(s)}$
    Calculate $\tilde{V}_{T+1}^{(s)}(\mathbf{X}_{T+1}^{(s)})$
  end if
end for
Save $\hat{\boldsymbol{\beta}}_{T+1}^{(1)}$ – a model where $\tilde{\mathbf{V}}_{T+1}$ is regressed on $\mathbf{X}_{T+1}$
Save $\hat{\boldsymbol{\beta}}_t^{(1)'} \leftarrow \hat{\boldsymbol{\beta}}_{T+1}^{(1)}$ $\forall t \in \{1,2,...,T\}$

4.3 Approximating the Value Function

The main difficulty in solving problem (1) is the high dimensional state space. A standard policy iteration algorithm requires enumerating the states, which is impractical for problem (1). Instead, we use a least squares regression to approximate the value function $V_t(\mathbf{X})$ as a function of the state $\mathbf{X}$. Instead of storing the value for each state $\mathbf{X}$, we only need to store the collection of regression coefficients. The regression coefficients can be estimated from a collection of randomly sampled states. In our algorithm, the sampled states are those visited when executing certain control policies. A crucial decision is the specification of the regression model. The simplest choice would be to approximate the value function as an affine function of the state. Indeed, the affine approximation is a popular approximation architecture in ADP. However, we do not expect such an approximation architecture to represent the cost well in our problem; in general, we do not expect the value function to be linear in the schedule.

In determining an appropriate set of variables, we seek to capture important "features" of the state variable; this is called "feature extraction" in Bertsekas and Tsitsiklis (1996). We create two sets of variables to approximate the value function. The first set captures the expected and total number of scheduled patients. To calculate the expected number of patients who show up at their appointment times, we take into account the cancellation and no-show probabilities of each scheduled patient.

We use second- and third-order polynomials of these variables in order to account for the possible non-linear relationship between these variables and the value function $V_t(\mathbf{X})$. The second set of variables describe the empty and overbooked appointment slots in $\mathbf{X}$. We record the number of empty slots, the maximum number of patients scheduled in a single slot and the variability in the number of patients assigned across slots. For example, consider a schedule with five slots where three patients are scheduled in slot 1 and one patient is scheduled in slots 2 through 5. Then, we have zero empty slots, a maximum of three patients scheduled in a single slot, and a variability index of 0.894, which corresponds to the standard deviation of 3, 1, 1, 1, and 1. The schedule-based predictors are especially important in ensuring that individuals are assigned to the available empty slots first before any overbooking is initiated. The list of predictor variables is given in Table 1. In total, there are 9 predictors. We use stepwise regression to select the variables to include in the final regression model.

Table 1 Model Predictors

Type	Predictor
Requests	Total # Patients
	Total # Patients^2
	Total # Patients^3
	E[Total # Patients]
	E[Total # Patients]^2
	E[Total # Patients]^3
Schedule	Empty Slots
	Max. Patients per Slot
	Overbooking Variability

4.4 Approximate Policy Iteration

In this section, we provide an overview of the API algorithm (see Algorithm 2). The procedure to generate arrivals and cancellations described in Section 4.1 is denoted by $f(u_t, \boldsymbol{\lambda_t}, \mathbf{q_t})$. $\mathbf{X}^{(s)}$ contains the schedule for sample path s and $\mathbf{X}_t^{(s)}$ is the schedule in period t for sample path s.

In the initialization step (Algorithm 1) for every sample path, we first set the schedule as empty; i.e. $\mathbf{X}_1^{(s)} \leftarrow 0$ (line 2). As noted in Section 4.2, we observe all arrivals until period T and schedule the patients based on Proposition 1 of Zacharias and Pinedo (2014) (lines 3 – 7). If the schedule is empty, its cost equals zero (lines 8 – 9); otherwise, we observe who shows up for their appointments and calculate the cost of the schedule (lines 10 – 14). The costs recorded for all S sample paths are regressed on the predictors listed in Table 1 and the regression parameters ($\beta_{T+1}^{(1)}$) are assigned to $\beta_t^{(1)}$, $\forall t$.

Following the initialization of the API, we repeat the loop in lines 2 – 43 of Algorithm 2 until the algorithm converges. We initialize the value function (line 4). For every sample path s we initialize

Algorithm 2 API algorithm

1: $i \leftarrow 1$
2: **repeat**
3: $i \leftarrow i+1$
4: $\mathbf{V}_t \leftarrow null$, $\forall t \in \{1,2,...,T+1\}$
5: **for** $s=1$ to S **do**
6: $\mathbf{X}_1^{(s)} \leftarrow \mathbf{0}$
7: $\mathbf{A} \leftarrow null$
8: **for** $t=1$ to T **do**
9: $u_t \leftarrow$ Uniform$[0,1]$
10: $u_t^* \leftarrow f(u_t, \boldsymbol{\lambda_t}, \mathbf{q_t})$ (see Section 4.1)
11: **for** $k=1$ to K **do**
12: **if** u_t^* = cancellation in class k **then**
13: $a_{tk} \leftarrow 0$
14: Update $\mathbf{X}_{t+1}^{(s)}$ based on u_t^* & $\mathbf{X}_t^{(s)}$
15: **end if**
16: **if** u_t^* = arrival in class k **then**
17: **if** $r_{tk} \geq \min_n\{\tilde{V}_{t+1}^{(s)}(\mathbf{X}_t^{(s)}+e_{nk}, \hat{\boldsymbol{\beta}}_{t+1}^{(i-1)}) - \tilde{V}_{t+1}^{(s)}(\mathbf{X}_t^{(s)}, \hat{\boldsymbol{\beta}}_{t+1}^{(i-1)})\}$ **then**
18: $a_{tk} \leftarrow 0$
19: Update $\mathbf{X}_{t+1}^{(s)}$ based on u_t^*, $\mathbf{X}_t^{(s)}$ & $n^* = \underset{n}{\operatorname{argmin}}\{\tilde{V}_{t+1}^{(s)}(\mathbf{X}_t^{(s)}+e_{nk}, \hat{\boldsymbol{\beta}}_{t+1}^{(i-1)}) - \tilde{V}_{t+1}^{(s)}(\mathbf{X}_t^{(s)}, \hat{\boldsymbol{\beta}}_{t+1}^{(i-1)})\}$
20: **else**
21: $a_{tk} \leftarrow 1$
22: $\mathbf{X}_{t+1}^{(s)} \leftarrow \mathbf{X}_t^{(s)}$
23: **end if**
24: **end if**
25: **if** u_t^* = neither arrival nor cancellation in class k **then**
26: $a_{tk} \leftarrow 0$
27: $\mathbf{X}_{t+1}^{(s)} \leftarrow \mathbf{X}_t^{(s)}$
28: **end if**
29: **end for**
30: **end for**
31: **if** $\mathbf{X}_{T+1}^{(s)} = \mathbf{0}$ **then**
32: $\tilde{V}_{T+1}^{(s)}(\mathbf{X}_{T+1}^{(s)}) \leftarrow 0$
33: **else**
34: Simulate no-shows based on $\mathbf{X}_{T+1}^{(s)}$ and $\mathbf{q}_{T+1}$
35: Serve the patients that show-up in $\mathbf{X}_{T+1}^{(s)}$
36: Calculate $\tilde{V}_{T+1}^{(s)}(\mathbf{X}_{T+1}^{(s)})$
37: **end if**
38: $\tilde{V}_t^{(s)}(\mathbf{X}_t^{(s)}) \leftarrow \sum_{i=t}^{T}\sum_k a_{ik}r_{ik} + \tilde{V}_{T+1}^{(s)}(\mathbf{X}_{T+1}^{(s)})$
39: **end for**
40: **for** $t=1$ to $T+1$ **do**
41: Save $\hat{\boldsymbol{\beta}}_t^{(i)}$ – a model where $\tilde{\mathbf{V}}_t$ is regressed on $\mathbf{X}_t$
42: $\hat{\boldsymbol{\beta}}_t^{(i)'} = (1-a_i) * \hat{\boldsymbol{\beta}}_t^{(i-1)'} + a_i * \hat{\boldsymbol{\beta}}_t^{(i)}$, where $a_i = \overline{a}/(\overline{a}+i-1), \overline{a} > 0$
43: **end for**
44: **until** Test if $\mu_{\tilde{\mathbf{V}}_{T+1}^{i}(\mathbf{x}_{T+1})} \neq \mu_{\tilde{\mathbf{V}}_{T+1}^{i-1}(\mathbf{x}_{T+1})}$, stop if p-value > 0.01

the schedule $\mathbf{X}_1^{(s)}$ and $\mathbf{A}$ (lines 6 – 7). $\mathbf{A}$ is the action matrix which contains the information on the arrival time and class of all rejected appointment requests. The loop in lines 8 – 30 shows how the algorithm makes scheduling decisions. If a patient cancels an appointment at time t (lines 12 – 15), we update our schedule ($\mathbf{X}_{t+1}^{(s)}$) by removing him or her from the schedule compiled between periods 1 and $t-1$. If a class-k patient requests an appointment (lines 16 – 24), he or she is assigned to the optimal time slot n^* when the expected cost increase in period t resulting from accepting the request does not exceed the rejection cost r_{tk} (lines 16 – 19). Otherwise the appointment request is rejected, and the rejection action is recorded (lines 20 – 23). The system stays the same when no event is observed (lines 25 – 28).

Once the $\mathbf{X}_{T+1}^{(s)}$ schedule is set, we observe its cost $\tilde{V}_{T+1}^{(s)}(\mathbf{X}_{T+1}^{(s)})$ (lines 31 – 37). To calculate

$\tilde{V}_t^{(s)}(\mathbf{X}_t^{(s)})$ for all t, we incorporate the cost of the rejected requests (line 38). Given a schedule $\mathbf{X}_t^{(s)}$ and its cost $\tilde{V}_t^{(s)}(\mathbf{X}_t^{(s)})$ for all S sample paths, we obtained the new regression parameters $(\beta_t^{(i)})$ for iteration i $(i \geq 2)$ for all t. The algorithm is terminated when the mean cost at time $T+1$ in iteration i is no longer significantly different at $\alpha = 0.01$ from the mean cost in period $T+1$ observed in iteration $i-1$ (line 44).

5. A Lower Bound and Benchmark Policies

This section introduces a lower bound on the total expected cost and several benchmark policies. The lower bound is based on the idea of information relaxation, where we assume full information (FI) on the arrivals and cancellations before scheduling patients in period $T+1$. We also introduce a no overbooking policy and two policies that accept all patient requests but differ in how overbooking is handled. These policies are compared with the one produced by our API algorithm in the numerical study.

Full Information (FI) Lower Bound: The procedure to compute the bound is summarized in Algorithm 3. For every sample path s, we observe all appointment and cancellation requests before a schedule is produced (lines 2 – 8). Therefore, we know the maximum number of individuals who could be seen at $T+1$ (M – line 9). Following Zacharias and Pinedo (2014), we rank patients ($rank_{tk}$ – line 10) based on their cost (r_{tk}) and no-show probabilities ($q_{T+1,k}$). We estimate the cost of scheduling the m ($m = \{N, N+1, ..., M\}$) highest $rank_{tk}$ patients. We (1) schedule the m patients (line 15) using Proposition 1 in Zacharias and Pinedo (2014), (2) simulate the patients who show up at $T+1$ given $q_{T+1,k}$ one thousand times (lines 16 – 19) and (3) calculate the average observed costs based on the simulated results (line 20). We pick m^* which has the lowest average observed simulation-based cost (line 23) and calculate $\tilde{V}_t^{(s)}$, $\forall t$ given m^* (lines 24 – 28).

PROPOSITION 2. *Assuming that the cost evaluation is exact, the cost produced by Algorithm 3 gives a lower bound on the total expected cost.*

The proof is immediate since Algorithm 3 utilizes additional information that is not available in the original problem. We make the assumption of exact cost evaluation since Algorithm 3 uses simulation to evaluate cost and naturally would introduce simulation error. We expect the simulation error to be small when the number of sample paths are relatively large.

No Overbooking (NO) Policy: Schedule patients on a first-come-first-serve basis. Do not overbook any appointment slots.

Accept All and Assign to Overtime Slots (ALL-I policy): Accept all appointment requests. Do not overbook appointment slots. See all overload patients during after hours.

Algorithm 3 Full Information Lower Bound

for $s = 1$ to S **do**
  $\mathbf{X}_1^{(s)} \leftarrow \mathbf{0}$
  $\mathbf{A} \longleftarrow null$
  for $t = 1$ to T **do**
    $u_t \leftarrow$ Uniform$[0,1]$
    $u_t^* \leftarrow f(u_t, \boldsymbol{\lambda_t}, \mathbf{q_t})$ (see Section 4.1)
  end for
  Update $\mathbf{X}_T^{(s)}$ based on $u_t^*, t \in \{1, T\}$, $\mathbf{X}_1^{(s)}$ & following Proposition 1 in Zacharias and Pinedo (2014)
  $M = \sum \mathbf{X}_T^{(s)}$
  Rank patient request $rank_{tk} = \frac{r_{tk}}{1 - q_{T+1,k}}$
  if $\mathbf{X}_{T+1}^{(s)} = \mathbf{0}$ **then**
    $\tilde{V}_{T+1}^{(s)}(\mathbf{X}_{T+1}^{(s)}) \leftarrow 0$
  else
    for $m = N$ to M **do**
      Accept the m highest rank $rank_{tk}$ patients $(\mathbf{X}_{T+1}^{(m,s)})$
      for $i = 1$ to 1000 **do**
        Simulate no-shows based on $\mathbf{X}_{T+1}^{(m,s)}$ and $\mathbf{q}_{T+1}$
        Calculate $\tilde{V}_{T+1}^{(i,s)}(\mathbf{X}_{T+1}^{(m,s)})$
      end for
      Calculate $E[\tilde{V}_{T+1}^{(s)}(\mathbf{X}_{T+1}^{(m,s)})] = \sum_{i=1}^{1000} \frac{\tilde{V}_{T+1}^{(i,s)}(\mathbf{X}_{T+1}^{(m,s)})}{1000}$
    end for
  end if
  $m^* = \underset{m}{\operatorname{argmin}} E[\tilde{V}_{T+1}^{(s)}(\mathbf{X}_{T+1}^{(m,s)})]$
  Accept the m^* highest rank $rank_{tk}$ patients $(\mathbf{X}_{T+1}^{(m^*,s)})$
  Save the information on the $M - m^*$ rejected patients in $\mathbf{A}$
  Simulate no-shows based on $\mathbf{X}_{T+1}^{(m^*,s)}$ and $\mathbf{q}_{T+1}$
  Calculate $\tilde{V}_{T+1}^{(s)}(\mathbf{X}_{T+1}^{(m^*,s)})$
  $\tilde{V}_t^{(s)}(\mathbf{X}_t^{(m^*,s)}) \leftarrow \sum_{i=t}^{T} \sum_k a_{ik} r_{ik} + \tilde{V}_{T+1}^{(s)}(\mathbf{X}_{T+1}^{(m^*,s)})$
end for

Accept All and Equally Disperse Across Slots (ALL-II policy): Accept all patient requests. If there are more requests than available slots, slots are evenly overbooked. For, example, if $M = 26$ and $N = 12$, 3 patients will be assigned to slot 1 and slot 7 and 2 individuals will be assigned to each remaining slot.

In addition, we also compare the API policy with a heuristic introduced in Zacharias and Pinedo (2014).

6. Parameter Estimation

This section estimates parameters for our dynamic programming model using a dataset obtained from a large public hospital in the US. The analysis in this section serves two purposes. First, we illustrate how the model parameters can be estimated using a real dataset. Second, the analysis allows us to construct realistic problem instances that are subsequently tested in Section 7. Section 6.1 describes the data. Section 6.2 estimates cancellation and no-show probabilities using survival analysis. Section 6.3 estimates arrival probabilities from the same dataset.

6.1 Data Description

The data contains information on 35,765 endocrinology appointments made by 12,593 individuals between January 1, 2011 and December 31, 2013. Tables 2 and 3 report summary statistics for

the appointment and patient data, respectively. 36% of the appointments are associated with individuals who are first time patients since January 1, 2011. The remaining appointments (64%) are for follow-up endocrinology visits. Most patients are seen on Monday (22.67%), Tuesday (25.30%) and Thursday (23.08%) and are almost exclusively allocated to 10 doctors and 2 clinics. 35.36% of the appointments are associated with individuals above the age of 65, evident in the proportion of visits covered under Medicare. On average, appointments are scheduled about 2 months in advance (average appointment lag $\approx$ 65). Some urgent visits are accommodated within a day (minimum appointment lag = 1), while preventative check-ups are sometimes scheduled more than a year in advance (maximum appointment lag = 375). Appointments take place between 7:00 AM and 5:00 PM. Most patients are female (78.92%) who speak English as their first language (97.58%), and are Caucasian (57.75%) and/or non-Hispanic (97.82%). A significant portion of the patients is married (45.14%) and/or unemployed (37.03%). The patient population is diverse with respect to age, including both infants (min age = 3) and centenarians (max age = 103). The average patient age is about 55. Patients tend to see the same provider (mean number of providers = 1.2470) and have the same insurance (mean number of insurances = 1.0932). On average, each patient requests 3.0295 appointments, where he or she attends 1.7684, cancels 0.7314 and no-shows for 0.5297 of them. Therefore, cancellations and no-shows account for a significant proportions of the appointments.

Table 2 Summary Statistics for the Appointment Data (35,765 observations)

A: Last Appointment = Arrival																		
	No	Yes																
Proportion	0.6419	0.3581																
B: Last Appointment = Cancellation																		
	No	Yes																
Proportion	0.8318	0.1682																
C: Last Appointment = No-Show																		
	No	Yes																
Proportion	0.8863	0.1137																
D: Days of the Week																		
	Monday	Tuesday	Wednesday	Thursday	Friday													
Proportion	0.2267	0.2530	0.1404	0.2308	0.1491													
E: Insurance Provider																		
	Aetna	Anthem BCBS	Cigna	Coventry	HealthKeepers	Indigent	Optima	United	Medicaid	Medicare	VA	Self-pay	Other					
Proportion	0.0298	0.2148	0.1170	0.0146	0.05114	0.02147	0.0117	0.0292	0.0560	0.3536	0.0573	0.0175	0.0259					
F: Medical Care Provider																		
	Doctor 1	Doctor 2	Doctor 3	Doctor 4	Doctor 5	Doctor 6	Doctor 7	Doctor 8	Doctor 9	Doctor 10	Nurse 1	Nurse 2	Nurse 3	Clinic 1	Clinic 2	Other		
Proportion	0.1363	0.1176	0.0577	0.0589	0.0492	0.0290	0.0170	0.0145	0.0088	0.0053	0.1547	0.0208	0.0106	0.1989	0.1201	0.0008		
G: Continuous Variables																		
	Mean	SD	Min	Max														
Appointment Lag	65.0298	72.3230	1	375														
Appointment Time	259.1735	136.9254	0 (7:00 AM)	600 (5:00 PM)														

Table 3 Summary Statistics for Patient Data (12,593 observations)

A: Gender						
	Male	Female				
Proportion	0.2108	0.7892				
B: Language						
	English	Other				
Proportion	0.9758	0.0242				
C: Race						
	Black	Caucasian	Other			
Proportion	0.3693	0.5775	0.0533			
D: Ethnicity						
	Hispanic	Not Hispanic	Other			
Proportion	0.01668	0.9782	0.0051			
E: Marital Status						
	Divorced	Married	Separated	Single	Widow	Other
Proportion	0.1201	0.4514	0.0313	0.3063	0.0858	0.0052
F: Employment Status						
	Full-Time	Part-Time	Self-Employed	Unemployed	Retired	Other
Proportion	0.2895	0.0524	0.0254	0.3703	0.2573	0.0052
G: Continuous Variables						
	Mean	SD	Min	Max		
Age	55.2915	16.8133	3	103		
No. of Providers	1.2470	0.5524	1	5		
No. of Insurances	1.0932	0.3363	1	5		
No. of Appointments	3.0295	3.2770	1	40		
No. of Arrivals	1.7684	2.1319	0	31		
No. of Cancellations	0.7314	1.2362	0	23		
No. of No-Shows	0.5297	1.0227	0	15		

6.2 Estimating Cancellation and No-show Probabilities

We examine five popular survival analysis models (see Table 4) to estimate the cancellation and no-show probabilities. They include a Cox proportional hazard model and four accelerated failure time (AFT) models (Klein and Moeschberger 2006). The four AFT models fit different distributions to the dependent variable: exponential, log-normal, log-logistic, and Weibull. We define a no-show as a cancellation made less than 24 hours before the actual appointment. The survival models consider cancellations (7,601) and no-shows (6,600) as events. Thus, there are 14,201 (= 7,601 + 6,600) events and 21,564 non-events (i.e. arrivals) in our survival analysis data set. The event time is set as the difference in days between the time the patient requests an appointment and cancels, does not show, or attends (whichever comes first).

We use the same predictors in all five models. They include (1) the lag between the appointment request and the actual visit, (2) the age of the patient when the request is made, (3) the gender, (4) whether the individual speaks English, the total number of previous appointments (5) scheduled, (6) attended, (7) canceled, the total number of appointment requests made in the last year, which are (8) scheduled, (9) attended, (10) canceled, whether a patient has (11) attended, (12) canceled or (13) not showed for his/her last appointment. We also incorporate a list of indicator variables pertaining to the request's (1) day of the week, (2) insurance provider, (3) medical care provider, as well as the patient's (4) race, (5) ethnicity, (6) marital status and (7) employment status. For more granular information on the seven categorical variables, refer to the groups in Table 3. The models are compared based on their AIC, BIC, and log-likelihood measures (Lambrecht and Tucker 2013, Sunder et al. 2017). The best model is a Weibull AFT model with the lowest AIC and BIC, and the highest log-likelihood.

Table 4 Survival Models

Predictor	Cox PH	Exponential	Log-Normal	Log-Logistic	Weibull
Appointment Lag	-0.0734^{***}	0.0110^{***}	0.0115^{***}	0.0111^{***}	0.0120^{***}
	(0.0008)	(0.0002)	(0.0001)	(0.0001)	(0.0001)
Age	-0.0029^{***}	0.0023^{***}	0.0008	0.0016^{**}	0.0009^{*}
	(0.0008)	(0.0008)	(0.0006)	(0.0007)	(0.0005)
Male	-0.0714^{***}	0.0829^{***}	0.0693^{***}	0.0855^{***}	0.0589^{***}
	(0.0207)	(0.0207)	(0.0145)	(0.0186)	(0.0125)
Language	0.2597^{***}	-0.2277^{***}	-0.1504^{***}	-0.1733^{***}	-0.1433^{***}
	(0.0689)	(0.0689)	(0.0510)	(0.0658)	(0.0417)
No. of Appointments	0.0784^{***}	-0.0768^{***}	-0.0455^{***}	-0.0564^{***}	-0.0445^{***}
	(0.0148)	(0.0147)	(0.0112)	(0.0147)	(0.0089)
No. of Arrivals	-0.1619^{***}	0.1538^{***}	0.0945^{***}	0.1224^{***}	0.0909^{***}
	(0.0187)	(0.0186)	(0.0138)	(0.0179)	(0.0113)
No. of Cancellations	−0.0191	0.0305	0.0100	0.0102	0.0156
	(0.0203)	(0.0202)	(0.0154)	(0.0199)	(0.0122)
No. of Appointments Last Year	0.0388^{*}	-0.0502^{**}	-0.0475^{***}	-0.0640^{***}	-0.0348^{***}
	(0.0219)	(0.0218)	(0.0165)	(0.0215)	(0.0132)
No. of Arrivals Last Year	−0.0059	0.0413	0.0492^{**}	0.0578^{**}	0.0352^{**}
	(0.0277)	(0.0277)	(0.0204)	(0.0264)	(0.0167)
No. of Cancellations Last Year	−0.0098	0.0295	0.0381^{*}	0.0566^{**}	0.0236
	(0.0297)	(0.0296)	(0.0222)	(0.0287)	(0.0179)
Last Appointment = Arrival	0.1182^{***}	-0.0994^{***}	-0.0598^{***}	-0.0762^{***}	-0.0604^{***}
	(0.0255)	(0.0256)	(0.0182)	(0.0229)	(0.0155)
Last Appointment = Cancellation	-0.1074^{***}	0.1048^{***}	0.0950^{***}	0.1291^{***}	0.0741^{***}
	(0.0298)	(0.0297)	(0.0213)	(0.0271)	(0.0180)
Last Appointment = No-show	0.0117	-0.0611^{*}	-0.0607^{**}	-0.0744^{**}	-0.0493^{**}
	(0.0336)	(0.0335)	(0.0247)	(0.0317)	(0.0202)
Days of the Week	Yes	Yes	Yes	Yes	Yes
Insurance Provider	Yes	Yes	Yes	Yes	Yes
Medical Care Provider	Yes	Yes	Yes	Yes	Yes
Race	Yes	Yes	Yes	Yes	Yes
Ethnicity	Yes	Yes	Yes	Yes	Yes
Marital Status	Yes	Yes	Yes	Yes	Yes
Employment Status	Yes	Yes	Yes	Yes	Yes
AIC	242,081.2	159,377.7	160,824.2	157,606.9	**155,499.1**
BIC	241,973.4	159,267.9	160,712.4	157,495.1	**155,387.3**
Log-likelihood	-120,982.6	-79,629.85	-80,352.1	-78,743.47	**-77,689.54**

$*p < 0.10$
$**p < 0.05$
$***p < 0.001$

After selecting the Weibull model, we examine its estimates and summarize the results in Table 5, which includes only the significant model predictors. According to the model, as the appointment lag increases, so does the patient's survival time (1.21% improvement per single day increase in the lag between a request and an actual appointment). Furthermore, older patients (0.09% increase) and males (6.07% increase) have higher survival times than younger patients and females. Patients whose first language is not English have significantly lower survival times (13.35% decrease). The more scheduled appointments patients attend overall (4.75% increase) as well as in the last 12 months (4.75% increase), the higher their survival times are. The opposite relationship is true when examining patients overall (2.85% decrease) as well as last year's (3.93% decrease) cancellations. Our model also suggests that patients who cancel their last appointment are more punctual (7.69% increase in survival time), while not showing up for an appointment further exacerbates one's punctuality problem (4.81% decrease in survival time).

We examined 10 variations on the Weibull AFT model (see Table 6). The models differ with respect to their predictors. Note that Model 6 in Table 6 is the same as the Weibull model in Table 4. Models 9 and 10 include the same independent variables as the ones in Model 6 plus one extra predictor, Appointment Time (Model 9) and the squared Appointment Lag (Model 10). While both models outperform Model 6, Model 10 is superior due to its low AIC, BIC and high log-likelihood.

Table 5 Weibull Model Significant Predictors' Interpretations

Predictors		Direction of Change	Percentage Change
Appointment Lag		↑	1.21
Age*		↑	0.09
Male		↑	6.07
Language		↓	13.35
No. of Arrivals		↑	4.75
No. of Cancellations		↓	2.85
No. of Arrivals Last Year		↑	4.75
No. of Cancellations Last Year		↓	3.93
Last Appointment = Arrival		↓	5.86
Last Appointment = Cancellation		↑	7.69
Last Appointment = No-Show		↓	4.81
Days of the Week (Baseline – Friday)	Monday	↑	9.25
	Tuesday	↑	5.03
	Wednesday	↑	5.34
	Thursday	↑	5.39
Insurance Provider (Baseline – Aetna)	Medicaid*	↓	6.45
	Self-pay	↓	34.98
	VA	↓	9.91
	Others	↓	9.2
Medical Provider (Baseline – Clinic 1)	Clinic 2	↓	7.66
	Doctor 1	↑	33.36
	Doctor 2	↑	28.27
	Doctor 3	↑	44.87
	Doctor 4	↑	64.91
	Doctor 5	↑	36.38
	Doctor 6	↑	13.25
	Doctor 8	↓	18.86
	Nurse 1	↑	28.4
Race (Baseline – African American)	Others	↑	6.61
Marital Status (Baseline – Divorced)	Widow*	↓	4.14
Employee Status (Baseline – Full-Time)	Not Employed*	↓	2.95
	Retired	↑	5.43
	Others	↓	14.48

$*0.05 < p < 0.10$ – borderline significant
↑ – increase
↓ – decrease

Therefore, we selected Model 10 as our final model. We do not provide a table similar to Table 5 for Model 10, as the coefficients in Models 6 and 10 are consistent with the exception of the Appointment Lag predictor, whose value changes in Model 10 due to the introduction of its squared transformation.

6.3 Estimating Arrival Rates

We estimate the arrival rates using the same predictors as the ones used in the survival analysis model. We obtain the frequency of requests coming from each unique patient-type (as described by the survival model predictors), which we call the unique patient frequency (UPF). As the dynamic programming model is week-day and provider specific, we aggregate the UPF values across these two predictors and refer to them as aggregate patient frequencies (APF). The ratio between the UPF and APF estimates the arrival rates used in the dynamic programming model. For example, on Monday, Doctor #1 has 2,159 unique patient types and there are two individuals with UPF #1, thus their arrival rate will be 2/2,159. Note that the UPF is conditional on the appointment lag. Thus, a patient of certain type can only request an appointment at a time consistent with his or her Appointment Lag value.

7. Numerical Study

In this section, we evaluate our model using a real world dataset. Section 7.1 compares the result of the API policy with the benchmark policies introduced in Section 5. Section 7.2 compares the

Table 6 Weibull Models

Predictor	Model 1	Model 2	Model 3	Model 4	Model 5	Model 6	Model 7	Model 8	Model 9	Model 10
Appointment Lag	0.0135***	0.0124***	0.0123***	0.0120***	0.0120***	0.0120***	0.0121***	0.0121***	0.0119***	164.2549***
	(0.0001)	(0.0001)	(0.0001)	(0.0001)	(0.0001)	(0.0001)	(0.0001)	(0.0001)	(0.0001)	(0.9713)
Squared Appointment Lag										−76.9768***
										(0.8759)
Appointment Times									−0.0004***	
									(0.0000)	
Age			0.0014***	0.0009**	0.0009*	0.0009*	0.0011**		0.0009*	0.0009**
			(0.0005)	(0.0005)	(0.0005)	(0.0005)	(0.0005)		(0.0005)	(0.0004)
Male			0.0653***	0.0571***	0.0581***	0.0589***	0.0577***	0.0623***	0.0552***	0.0298***
			(0.0125)	(0.0125)	(0.0125)	(0.0125)	(0.0125)	(0.0123)	(0.0125)	(0.0100)
Language			−0.1524***	−0.1417***	−0.1423***	−0.1433***	−0.1664***	−0.1612***	−0.1391***	−0.1245***
			(0.0419)	(0.0417)	(0.0417)	(0.0417)	(0.0395)	(0.0395)	(0.0416)	(0.0330)
No. of Appointments				−0.0697***	−0.0450***	−0.0445***	−0.0344***	−0.0346***	−0.0434***	−0.0373***
				(0.0047)	(0.0089)	(0.0089)	(0.0052)	(0.0052)	(0.0089)	(0.0071)
No. of Arrivals				0.1101***	0.0913***	0.0909***	0.0807***	0.0813***	0.0893***	0.0721***
				(0.0060)	(0.0113)	(0.0113)	(0.0087)	(0.0087)	(0.0113)	(0.0090)
No. of Cancellations				0.0482***	0.0132	0.0156			0.0157	0.0115
				(0.0064)	(0.0122)	(0.0122)			(0.0122)	(0.0099)
No. of Appointments Last Year					−0.0433***	−0.0348***	−0.0191**	−0.0193**	−0.0332**	−0.0138
					(0.0125)	(0.0132)	(0.0076)	(0.0076)	(0.0131)	(0.0104)
No. of Arrivals Last Year					0.0303*	0.0352**	0.0199	0.0197	0.0329**	0.0024
					(0.0158)	(0.0167)	(0.0130)	(0.0129)	(0.0167)	(0.0132)
No. of Cancellations Last Year					0.0578***	0.0236			0.0221	0.0046
					(0.0168)	(0.0179)			(0.0179)	(0.0143)
Last Appointment = Arrival						−0.0604***	−0.0614***	−0.0608***	−0.0669***	−0.0669***
						(0.0155)	(0.0155)	(0.0155)	(0.0155)	(0.0122)
Last Appointment = Cancellation						0.0741***	0.0903***	0.0917***	0.0710***	0.0729***
						(0.0180)	(0.0174)	(0.0174)	(0.0179)	(0.0143)
Last Appointment = No-show						−0.0493**	−0.0820***	−0.0825***	−0.0482**	−0.0133
						(0.0202)	(0.0185)	(0.0185)	(0.0202)	(0.0161)
Days of the Week	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Insurance Provider	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Medical Care Provider	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Race	No	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Ethnicity	No	No	Yes	Yes	Yes	Yes	No	No	Yes	Yes
Marital Status	No	No	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes
Employment Status	No	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
AIC	157,357.1	156,036.8	155,907.7	155,564.7	155,552.8	155,499.1	155,513.3	155,522.9	155,398.3	**150,455.5**
BIC	157,353.3	155,975.9	155,813.6	155,464.6	155,446.9	155,387.3	155,409.3	155,430.7	155,284.5	**150,341.7**
Log-likelihood	-78,675.57	-77,984.42	-77,902.87	-77,728.33	-77,719.4	-77,689.54	-77,700.63	-77,711.44	-77,638.15	**-75,166.76**

$*p < 0.10$
$**p < 0.05$
$***p < 0.001$

performance of the API policy to the heuristic introduced in Zacharias and Pinedo (2014).

7.1 The API Policy vs. Benchmark Policies

We compare the API policy to the benchmark policies with respect to the most popular doctor on her busiest day: Doctor #1 on Monday in Table 7. We focus on the 120-day time window as 84.67% of all 35,765 appointment requests are within 120 days. 2,160 of the 2,725 patients were seen by Doctor #1 within 120 days of an appointment request. Of the 2,160 patients, 2,159 represented unique patient classes as defined by the predictors in the survival analysis model in Table 6 because we had 2 patients with the exact same characteristics as defined by these predictors. Thus, our model accounted for 2,159 patient classes with appointment requests arriving over 120 days. All accepted appointments are scheduled on day 121.

We simulate $S = 11,000$ sample paths with patient requests. We use 1,000 sample paths to train the API algorithm and evaluate it on the remaining 10,000. All sample paths are simulated using the arrival rates and cancellation probabilities discussed in Section 6. We also evaluate all benchmark

Table 7 Medical Care Provider & Work-day Frequencies

Provider	Monday	Tuesday	Wednesday	Thursday	Friday	Total
Doctor #1	**2,725**	1,541	36	517	54	4,873
Doctor #2	350	1,228	116	1,312	1,199	4,205
Doctor #3	17	13	5	8	2,022	2,065
Doctor #4	655	728	55	662	5	2,105
Doctor #5	971	500	129	27	134	1,761
Doctor #6	0	0	341	696	0	1,037
Doctor #7	17	11	9	565	6	608
Doctor #8	102	155	121	141	0	519
Doctor #9	98	75	35	0	106	314
Doctor #10	88	47	0	0	53	188
Nurse #1	218	1,806	1,750	1,663	95	5,532
Nurse #2	238	102	0	230	174	744
Nurse #3	120	249	3	6	0	378
Clinic #1	1,536	1,607	1,557	1,622	792	7,114
Clinic #2	946	985	863	807	693	4,294
Other	28	0	0	0	0	28

policies on the same 10,000 sample paths.

To evaluate how the API policy performs across different system loads, we vary the available appointment slots at time $T+1$, such that $N \in \{4, 6, 8, 10, 12, 14, 16\}$. The corresponding system loads are 2.57, 1.72, 1.29, 1.03, 0.86, 0.74 and 0.64, respectively. As noted in line 42 of Algorithm 2, we need to specify the value of $\overline{a}$. We examine the API algorithm for $\overline{a} \in \{5, 10, 20, 30, 40\}$ and determined that $\overline{a} = 20$ provides the best overall performance.

Table 8 API Model Results

N	Summary	FI	Policy				API Iteration											
	Statistics	Bound	ALL-I	ALL-II	NO	Best API	1	2	3	4	5	6	7	8	9	10	11	12
4	mean	3.60	6.76	9.29	6.63	4.02	6.60	6.92	6.73	5.56	5.35	5.61	4.02	4.23	4.60	4.31	4.44	4.30
	median	3.49	6.50	7.70	6.58	3.83	6.51	5.94	6.66	5.19	5.27	5.51	3.83	4.10	4.47	4.16	4.30	4.18
	max	11.35	18.00	44.70	14.30	13.83	16.63	43.32	14.47	26.67	14.23	18.44	15.02	14.45	13.83	14.53	15.13	14.53
	s.d.	1.73	2.46	5.94	2.07	1.74	2.05	5.01	1.82	3.44	1.74	2.25	1.94	1.85	1.86	1.96	1.92	1.85
6	mean	2.47	5.76	6.88	5.94	2.58	5.26	4.20	6.07	3.74	3.40	3.59	2.58					
	median	2.33	5.50	5.60	5.88	2.44	5.19	3.36	6.00	3.00	3.29	3.42	2.44					
	max	10.24	17.00	39.50	13.42	12.17	14.80	33.23	16.37	23.20	12.17	15.37	13.26					
	s.d.	1.65	2.46	5.39	2.11	1.63	2.06	4.36	1.80	3.25	1.67	1.94	1.63					
8	mean	1.49	4.79	4.73	4.77	2.38	4.00	2.93	3.16	2.38								
	median	1.19	4.50	3.40	4.72	2.04	3.88	2.04	3.05	2.26								
	max	9.01	16.00	34.70	12.20	12.86	13.80	20.61	13.69	12.86								
	s.d.	1.45	2.41	4.42	2.02	1.53	1.98	2.95	1.66	1.53								
10	mean	0.73	3.92	3.56	3.83	1.66	3.26	2.15	2.96	2.02	1.66	1.69	1.70					
	median	0.00	3.50	2.50	3.71	1.19	3.02	1.19	2.83	1.88	1.53	1.56	1.58					
	max	7.63	15.00	34.00	10.75	10.71	11.85	26.98	11.68	11.57	11.19	10.88	10.71					
	s.d.	1.11	2.27	3.36	1.89	1.39	1.84	2.97	1.62	1.50	1.46	1.40	1.39					
12	mean	0.28	3.26	2.98	3.19	1.38	2.82	1.96	2.25	1.38								
	median	0.00	3.00	2.40	3.00	1.25	2.60	1.40	2.13	1.25								
	max	6.42	14.00	24.70	9.75	10.65	11.22	27.05	11.33	10.65								
	s.d.	0.71	2.05	2.46	1.81	1.47	1.64	2.42	1.47	1.50								
14	mean	0.08	2.86	2.74	2.82	1.85	2.70	2.21	1.85									
	median	0.00	3.00	2.20	2.90	1.68	2.49	1.81	1.68									
	max	5.70	13.00	22.10	8.70	10.82	10.82	23.46	12.37									
	s.d.	0.38	1.84	1.93	1.69	1.52	1.56	2.05	1.52									
16	mean	0.02	2.69	2.66	2.68	1.26	2.68	2.11	1.83	1.26	1.26	1.48						
	median	0.00	2.00	2.00	2.00	1.07	2.47	1.81	1.60	1.07	1.11	1.37						
	max	4.05	12.00	17.90	10.92	9.99	9.99	25.01	11.90	11.68	10.32	10.36						
	s.d.	0.18	1.71	1.70	1.66	1.34	1.55	1.75	1.57	1.34	1.39	1.37						

Table 8 shows the costs for different policies. Based on Table 8, the API algorithm converges within at most 12 iterations. It outperforms the NO and both ALL policies, with the exception of the maximum summary statistics for $N = 12$ and 14, where the NO policy has slightly better results. When compared to the FI lower bound, the API policy produces comparable results when the demand for care surpasses the supply ($N = 4, 6, 8, 10$). The performance of the API policy deteriorates as the system load decreases. The gap from the bound is most pronounced when the system load is below 1 ($N = 12, 14, 16$). Note that the FI lower bound is very optimistic as it assumes

full knowledge of arrivals and cancellations before any appointment decisions are made. In contrast, the API policy can book an appointment request, which is subsequently cancelled close to period $T+1$. We point out that 15.33% of the appointment requests in our data are made more than 120 days in advance (1.03% of them over 1 year in advance). Thus, it is unlikely that the true system load in the hospital under study is below 1.

7.2 The API Policy vs. Zacharias and Pinedo (2014) Heuristic

In this section, we compare the performance of the API policy against the heuristic presented in Zacharias and Pinedo (2014). We follow the settings described in Section 6.2 of Zacharias and Pinedo (2014) and simulate 11,000 sample paths. Note that the instances involve no-shows but no cancellations. We use 1,000 of the sample paths to train the API algorithm. The remaining 10,000 are used to evaluate the performance of the two policies. We compare the two approaches across all settings identified in Table 4 on page 12 of Zacharias and Pinedo (2014).

We should note that when comparing our model to the results reported in Zacharias and Pinedo (2014) we include 14 predictors not included in Table 1. To evaluate their heuristic, Zacharias and Pinedo (2014) assume that patients can be grouped into 4 classes. Thus, it is manageable to include the Total # of Patients from class k and E[Total # of Patients from class k], where $k \in \{1,2,3,4\}$. Note that in the numerical study based on real appointment data (see §7.1) we treat each patient as belonging to his/her own class, resulting in 2,159 unique patient types. This is why we do not use separate variables for each of our patient classes. Because the evaluation performed by Zacharias and Pinedo focuses on scheduling 14 patients over 12 appointment slots, we introduce six indicator variables that determine if the distance between the two overbooked appointment slots (if two slots end up being overbooked) is 0 (i.e. the overbooked slots neighbor one another), 1, 2, 3, 4, or more than 4. We use binary variables here because patients do not cancel (cancellation probability is 0).

We report the percentage difference between the mean cost of the heuristic policy vs. the mean cost of the API policy $((\mu_{V_{API}} - \mu_{V_{huristic}})/\mu_{V_{huristic}})$ across the 10,000 sample paths. We also perform a two-sided paired means hypothesis test to determine if the difference between the two policies is significant. A negative difference indicates that the API policy outperforms the heuristic.

The results from the 96 settings are reported in Table 9. The Zacharias and Pinedo (2014) heuristic dominates in only 2 settings (*italicized*). Both settings were associated with patient populations which had fairly homogeneous no-show probabilities (i.e. 0. and 0.05). There was no significant difference in 20 settings, which were once again associated with relatively high patient homogeneity – i.e., $\Delta q \in \{0, 0.05, 0.1, 0.15, 0.2\}$. The API policy significantly outperform the Zacharias and Pinedo (2014) heuristic in the remaining 74 settings (in **bold**).

Table 9 API Model Results

Δw \ Δq	0	0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5	0.55
0	−0.33	**−2.41*****	**−3.51*****	**−3.32*****	**−3.66*****	**−3.64*****	**−4.33*****	**−4.54*****	**−6.20*****	**−6.88*****	**−6.12*****	**−6.82*****
0.05	−0.63	**−1.96****	**−2.99*****	**−3.01*****	**−3.22*****	**−4.30*****	**−4.69*****	**−4.83*****	**−6.36*****	**−7.07*****	**−4.52*****	**−5.73*****
0.1	−0.94	**−1.37***	**−2.57*****	−0.91	**−2.97*****	**−4.00*****	**−4.46*****	**−4.96*****	**−6.51*****	**−7.13*****	**−6.50*****	**−8.03*****
0.15	*1.96****	−0.49	**−2.02****	**−1.68***	**−2.41*****	**−3.30*****	**−3.84*****	**−4.17*****	**−6.53*****	**−7.07*****	**−6.12*****	**−7.76*****
0.2	0.15	0.37	−1.32	−1.03	**−2.26****	**−3.39*****	**−4.09*****	**−4.35*****	**−6.85*****	**−7.33*****	**−6.52*****	**−8.02*****
0.25	**−1.33***	0.71	−0.90	−0.43	−1.12	**−2.82*****	**−3.72*****	**−3.84*****	**−6.50*****	**−7.05*****	**−6.45*****	**−8.19*****
0.3	**−2.16*****	1.15	−1.24	−0.85	−1.37	**−3.12*****	**−4.39*****	**−4.83*****	**−6.94*****	**−7.16*****	**−7.18*****	**−8.98*****
0.35	**−2.36*****	*1.87**	−0.17	0.44	−0.80	**−2.44****	**−3.17*****	**−3.84*****	**−7.43*****	**−7.05*****	**−6.33*****	**−8.48*****

$*p < 0.10$
$**p < 0.05$
$***p < 0.001$

8. Conclusion

We study the dynamic appointment scheduling problem with cancellations and overbooking. The problem is formulated as a finite-horizon stochastic dynamic program. The model has high dimensional state space and is not amenable to exact solution methods. We propose an API algorithm together with a value function approximation. We validate the model with data from a large public hospital and compare our solution methods with several benchmarks. Our numerical results indicate that the proposed method is competitive against several computational benchmarks.

The proposed solution method has several essential strengths. First, we use a least-squares regression to estimate the value function. Our approximation architecture is nonlinear, since the regression model incorporates higher order polynomial terms. In contrast, with LP-based ADP, the polynomial terms in the value function approximation accounts for non-linearity and make it necessary to solve large scale nonlinear optimization problems that are not well structured (e.g., lack of convexity). Second, the method is simulation-based, and therefore can incorporate rather complex system dynamics. This latter feature makes the proposed approach particularly appealing for practical implementation, where complex customer behavior might arise and necessitate more detailed modeling. Third, the solution method can be used to produce control policies offline, which can then be used for online control. Therefore, the solution time is not a major obstacle.

The test instances considered in our numerical study are produced based on a dataset from a large hospital. We use a Weibull AFT model to estimate patients' cancellation and no-show behavior over time. Our empirical approach is fully personalizable as patients are not clustered into classes. Taken together, the empirical part of our work constitutes a viable approach to estimate problem parameters from data.

The value function approximation in our work is based on least-squares regression. An alternative is to consider other reasonable choices for the approximation architecture, such as regression trees and neural networks. Using a more flexible approximation architecture, such as trees or neural networks, is likely to increase the computational time. Such an increase in computational time would be worthwhile if the solution quality increases substantially. We leave such investigation to future research.

References

Adelman, D. (2003). Price-directed replenishment of subsets: Methodology and its application to inventory routing. *Manufacturing & Service Operations Management*, 5(4):348–371.

Adelman, D. (2004). A price-directed approach to stochastic inventory/routing. *Operations Research*, 52(4):499–514.

Ahmadi-Javid, A., Jalali, Z., and Klassen, K. J. (2017). Outpatient appointment systems in healthcare: A review of optimization studies. *European Journal of Operational Research*, 258(1):3–34.

Alagoz, O., Hsu, H., Schaefer, A. J., and Roberts, M. S. (2010). Markov decision processes: a tool for sequential decision making under uncertainty. *Medical Decision Making*, 30(4):474–483.

Barz, C. and Rajaram, K. (2013). Elective patient admission and scheduling under multiple resource constraints. Working paper, Anderson School of Management, University of California, Los Angeles.

Bertsekas, D. P. (2011). Approximate dynamic programming. Working Paper (book chapter draft), MIT.

Bertsekas, D. P. and Tsitsiklis, J. N. (1996). *Neuro-Dynamic Programming*. Athena Scientific, Belmont, MA.

Cayirli, T. and Veral, E. (2003). Outpatient scheduling in health care: a review of literature. *Production and Operations Management*, 12(4):519–549.

de Farias, D. and Van Roy, B. (2004). On constraint sampling in the linear programming approach to approximate dynamic programming. *Mathematics of Operations Research*, 29(3):462–478.

de Farias, D. P. and Van Roy, B. (2006). A cost-shaping linear program for average-cost approximate dynamic programming with performance guarantees. *Mathematics of Operations Research*, 31(3):597–620.

Feldman, J., Liu, N., Topaloglu, H., and Ziya, S. (2012). Appointment scheduling under patient preference and no-show behavior. Working paper, Columbia University.

Gocgun, Y., Bresnahan, B. W., Ghate, A., and Gunn, M. L. (2011). A Markov decision process approach to multi-category patient scheduling in a diagnostic facility. *Artificial intelligence in medicine*, 53(2):73–81.

Gocgun, Y. and Puterman, M. L. (2014). Dynamic scheduling with due dates and time windows: an application to chemotherapy patient appointment booking. *Health care management science*, 17(1):60–76.

Grant, B., Gurvich, I., Mutharasan, R. K., and Van Mieghem, J. (2017). Optimal dynamic appointment scheduling of base and surge capacity. Working paper, Northwestern University.

Gupta, D. and Denton, B. (2008). Appointment scheduling in health care: Challenges and opportunities. *IIE transactions*, 40(9):800–819.

Gupta, D. and Wang, L. (2008). Revenue management for a primary-care clinic in the presence of patient choice. *Operations Research*, 56(3):576–592.

Hassin, R. and Mendel, S. (2008). Scheduling arrivals to queues: A single-server model with no-shows. *Management Science*, 54(3):565–572.

Kaandorp, G. C. and Koole, G. (2007). Optimal outpatient appointment scheduling. *Health Care Management Science*, 10(3):217–229.

Klassen, K. J. and Yoogalingam, R. (2009). Improving performance in outpatient appointment services with a simulation optimization approach. *Production and Operations Management*, 18(4):447–458.

Klein, J. P. and Moeschberger, M. L. (2006). *Survival analysis: techniques for censored and truncated data*. Springer Science & Business Media.

Koch, S. (2017). Least squares approximate policy iteration for learning bid prices in choice-based revenue management. *Computers & Operations Research*, 77:240–253.

Kolesar, P. (1970). A markovian model for hospital admission scheduling. *Management Science*, 16(6):B–384.

LaGanga, L. R. and Lawrence, S. R. (2007). Clinic overbooking to improve patient access and increase provider productivity. *Decision Sciences*, 38(2):251–276.

LaGanga, L. R. and Lawrence, S. R. (2012). Appointment overbooking in health care clinics to improve patient service and clinic performance. *Production and Operations Management*, 21(5):874–888.

Lambrecht, A. and Tucker, C. (2013). When does retargeting work? information specificity in online advertising. *Journal of Marketing Research*, 50(5):561–576.

Liu, N., Ziya, S., and Kulkarni, V. G. (2010). Dynamic scheduling of outpatient appointments under patient no-shows and cancellations. *Manufacturing & Service Operations Management*, 12(2):347–364.

Mondschein, S. and Weintraub, G. (2003). Appointment policies in service operations: A critical analysis of the economic framework. *Production and Operations Management*, 12(2):266–286.

Patrick, J. (2012). A markov decision model for determining optimal outpatient scheduling. *Health care management science*, 15(2):91–102.

Patrick, J., Puterman, M., and Queyranne, M. (2008). Dynamic multi-priority patient scheduling for a diagnostic resource. *Operations Research*, 56(6):1507–1525.

Powell, W. (2007). *Approximate Dynamic Programming: Solving the Curses of Dimensionality*. Wiley-Interscience, Hoboken, New Jersey.

Puterman, M. L. (1994). *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, New York.

Robinson, L. W. and Chen, R. R. (2010). A comparison of traditional and open-access policies for appointment scheduling. *Manufacturing & Service Operations Management*, 12(2):330–346.

Samiedaluie, S., Kucukyazici, B., Verter, V., and Zhang, D. (2014). Managing patient admissions in a neurology ward. Forthcoming, *Operations Research*.

Samorani, M. and LaGanga, L. (2013). Outpatient appointment scheduling given individual day-dependent no-show predictions. Working paper, School of Business, University of Alberta.

Sauré, A., Patrick, J., and Puterman, M. L. (2015). Simulation-based approximate policy iteration with generalized logistic functions. *INFORMS Journal on Computing*, 27(3):579–595.

Sauré, A., Patrick, J., Tyldesley, S., and Puterman, M. L. (2012). Dynamic multi-appointment patient scheduling for radiation therapy. *European Journal of Operational Research*, 223(2):573–584.

Schütz, H.-J. and Kolisch, R. (2012). Approximate dynamic programming for capacity allocation in the service industry. *European Journal of Operational Research*, 218(1):239–250.

Schweitzer, P. J. and Seidmann, A. (1985). Generalized polynomial approximations in Markovian decision processes. *Journal of Mathematical Analysis and Applications*, 110(2):568–582.

Subramanian, J., Stidham, S., and Lautenbacher, C. J. (1999). Airline yield management with overbooking, cancellations, and no-shows. *Transportation Science*, 33:147–168.

Sunder, S., Kumar, V., Goreczny, A., and Maurer, T. (2017). Why do salespeople quit? an empirical examination of own and peer effects on salesperson turnover behavior. *Journal of Marketing Research*, 54(3):381–397.

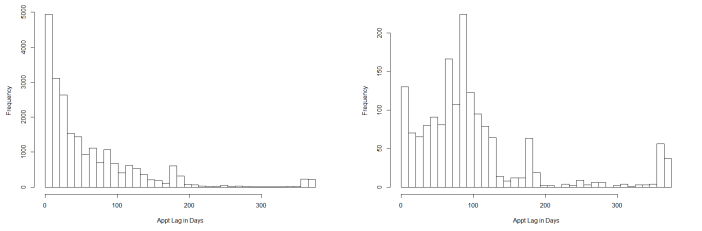
Wang, W.-Y. and Gupta, D. (2011). Adaptive appointment systems with patient preferences. *Manufacturing & Service Operations Management*, 13(3):373–389.

Zacharias, C. and Pinedo, M. (2014). Appointment scheduling with no-shows and overbooking. *Production and Operations Management*, 23(5):788–801.

Appendix A: Summary of Notations

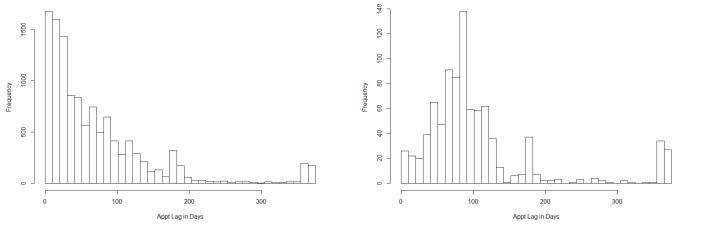
T	The number of periods ($t \in \{1, 2, ..., T\}$).
K	The number of patient classes ($k \in \{1, 2, ..., K\}$).
N	The number of appointment slots in period $T+1$.
S	The number of sample paths generated.
λ_{tk}	The arrival rate of a class-k patient in period t.
q_{tk}	The cancellation probability of a class-k patient in period t.
$q_{T+1,k}$	The no-show probability of a class-k patient.
r_{tk}	The rejection cost of a class-k patient in period t.
α	The unit waiting cost.
β	The overtime premium for doctors.

Appendix B: Descriptive Statistics



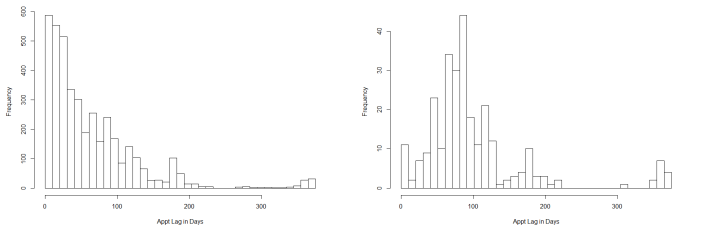
(a) All Data (b) Doctor #1 on Monday

Figure B1 **Arrivals Given Request Lag Time**



(a) All Data (b) Doctor #1 on Monday

Figure B2 **Cancelled Appointments Given Request Lag Time**



(a) All Data (b) Doctor #1 on Monday

Figure B3 **No-Shows Given Request Lag Time**

Table B1 Correlation Categorical Predictors

	Male	Language	Last Appointment = Arrival	Last Appointment = Cancellation	Last Appointment = No-Show	Day of the Week	Insurance Provider	Medical Care Provider	Race	Ethnicity	Marital Status	Employment Status
Male		−0.0076	0.0824***	−0.0254***	0.0075	0.009*	0.0043	0.0480***	0.0183***	0.0129**	−0.0334***	0.0105**
Language	−0.0076		−0.0457***	−0.0126**	-0.0023	0.0029	0.0614***	−0.0472***	−0.0200***	−0.0715***	0.0093*	0.0033
Last Appointment = Arrival	0.0824***	−0.0457***		−0.3358***	−0.2675***	−0.0390***	−0.0345***	0.1554***	0.0138**	-0.0067	−0.0158**	-0.0015
Last Appointment = Cancellation	−0.0254***	−0.0126**	−0.3358***		−0.1611***	0.0047	−0.0212***	0.0074	0.0321***	0.0021	−0.0097*	−0.0128**
Last Appointment = No-Show	0.0075	-0.0023	−0.2675***	−0.1611***		0.0135**	0.0568***	0.0058	−0.0909***	-0.0074	0.0289***	0.0067
Day of the Week	0.009*	0.0029	−0.0390***	0.0047	0.0135**		0.0508***	0.1459***	−0.0597***	−0.0074	0.0263***	0.0465***
Insurance Provider	0.0043	0.0614***	−0.0345***	−0.0212***	0.0568***	0.0508***		−0.0731***	−0.1829***	−0.0147**	0.0705***	0.3345***
Medical Care Provider	0.0480***	−0.0472***	0.1554***	0.0074	0.0058	0.1459***	−0.0731***		0.0287***	0.0060	0.0178***	−0.0205***
Race	0.0183***	−0.0200***	0.0138**	0.0321***	−0.0909***	−0.0597***	−0.1829***	0.0287***		−0.0033	−0.1681***	−0.0183***
Ethnicity	0.0129**	−0.0715***	-0.0067	0.0021	-0.0074	−0.0074	−0.0147**	0.0060	−0.0033		0.0029	0.0163**
Marital Status	−0.0334***	0.0093*	−0.0158**	−0.0097*	0.0289***	0.0263***	0.0705***	0.0178***	−0.1681***	0.0029		0.0259***
Employment Status	0.0105**	0.0033	-0.0015	−0.0128**	0.0067	0.0465***	0.3345***	−0.0205***	−0.0183***	0.0163**	0.0259***	

$*p < 0.10$
$**p < 0.05$
$***p < 0.001$

Table B2 Correlation Continuous Predictors

	Appointment Lag	Appointment Times	Age	No. of Appointments	No. of Arrivals	No. of Cancellations	No. of Appointment Last Year	No. of Arrivals Last Year	No. of Cancellations Last Year
Appointment Lag		−0.18***	0.06***	0.10***	0.15***	0.04***	0.03***	0.12***	−0.01*
Appointment Times	−0.18***		−0.06***	−0.01**	−0.04***	0.00	0.00	−0.04***	0.04
Age	0.06***	−0.06***		−0.04***	−0.03***	0.00	−0.07***	−0.06***	−0.01**
No. of Appointments	0.10***	−0.01**	−0.04***		0.90***	0.72***	0.83***	0.72***	0.55***
No. of Arrivals	0.15***	−0.04***	−0.03***	0.90***		0.46***	0.73***	0.83***	0.31***
No. of Cancellations	0.04***	0.00	0.00	0.72***	0.46***		0.63***	0.34***	0.84***
No. of Appointment Last Year	0.03***	0.00	−0.07***	0.83***	0.73***	0.63***		0.84***	0.68***
No. of Arrivals Last Year	0.12***	−0.04***	−0.06***	0.72***	0.83***	0.34***	0.84***		0.31***
No. of Cancellations Last Year	−0.01*	0.04	−0.01**	0.55***	0.31***	0.84***	0.68***	0.31***	

$*p < 0.10$
$**p < 0.05$
$***p < 0.001$