# A PREFERENCE-BASED APPOINTMENT SCHEDULING PROBLEM WITH MULTIPLE PATIENT TYPES

Feray TUNÇALP[1*], Lerzan ÖRMECİ[2]

[1] Koç University, Faculty of Engineering, Department of Industrial Engineering, İstanbul, Turkey
ORCID No : http://orcid.org/0000-0001-7542-1895
[2] Koç University, Faculty of Engineering, Department of Industrial Engineering, İstanbul, Turkey
ORCID No : http://orcid.org/0000-0003-3575-8674

**Keywords**

*Healthcare, appointment scheduling, patient preferences, Markov Decision Processes, Approximate Dynamic Programming*

**Abstract**

*This paper focuses on the appointment scheduling mechanism of a physician or a diagnostic resource in a healthcare facility. Multiple patient types with different revenues use the facility. The facility observes the number of appointment requests arriving from each patient type at the beginning of each day. It decides on how to allocate available appointment slots to these appointment requests. Patients prefer a day in the booking horizon with a specific probability and they have only one preference. Patients are either given an appointment for their preferred days or their appointment requests are rejected. The facility wants to keep the rejection costs at a certain level, while maximizing its revenues. This process is modeled with a discrete time and constrained Markov Decision Process to maximize the infinite-horizon expected discounted revenue. The constraint guarantees that the infinite-horizon expected discounted rejection cost is below a specific threshold. We have proved that the optimal policy is a randomized booking limit policy. To solve the model, we have implemented Temporal Difference (TD) Learning Algorithm, which is a well-known Approximate Dynamic Programming (ADP) method. We have compared the ADP results with other heuristics numerically.*

# ÇOK TİPLİ HASTALAR İÇİN TERCİHLERİ BAZ ALAN BİR RANDEVU ÇİZELGELEME PROBLEMİ

**Anahtar Kelimeler**

*Sağlık hizmeti, randevu çizelgeleme, hasta tercihleri, Markov Karar Süreçleri, Yaklaşık Dinamik Programlama*

**Öz**

*Bu makale, bir sağlık tesisindeki bir doktor ya da tanı cihazının randevu planlama mekanizmasına odaklanmaktadır. Bu tesisi, getirileri birbirinden farklı olan birden çok hasta tipi kullanmaktadır. Tesis, her hasta tipinden gelen randevu isteklerini her günün başında gözlemlemektedir. Müsait randevu saatlerini bu randevu isteklerine nasıl tahsis edeceğine karar vermektedir. Hastalar belli bir olasılıkla rezervasyon dönemindeki bir günü tercih etmektedirler ve sadece bir tercihleri vardır. Hastalara ya tercih ettiği güne bir randevu verilmektedir ya da randevu istekleri reddedilmektedir. Tesis, getirilerini maksimize ederken reddedilme maliyetlerini belli bir seviyede tutmak istemektedir. Bu süreç, sonsuz zamanlı beklenen indirgenmiş karı maksimize etmek için ayrık zamanlı ve kısıtlı Markov Karar Süreci ile modellenmektedir. Kısıt, sonsuz zamanlı beklenen indirgenmiş reddedilme maliyetlerinin belli bir eşik değerinin altında olmasını garanti etmektedir. En iyi politikanın rassallaştırılmış bir rezervasyon limiti politikasının olduğunu gösterdik. Modeli çözmek için iyi bilinen bir "Yaklaşık Dinamik Programlama" metodu olan "Geçici Farklarla Öğrenme Algoritmasını" uyguladık. "Yaklaşık Dinamik Programlama" sonuçlarını diğer buluşsal yöntemlerle sayısal olarak karşılaştırdık.*

| | | | |
|---|---|---|---|
| Araştırma Makalesi | | Research Article | |
| Başvuru Tarihi | : 09.07.2019 | Submission Date | : 09.07.2019 |
| Kabul Tarihi | : 03.10.2019 | Accepted Date | : 03.10.2019 |

[*] Correspondin author; e-mail: ftuncalp@ku.edu.tr

## 1. Introduction

Improving the healthcare systems has been one of the most essential aims of the developed countries. According to OECD health statistics, healthcare expenditures constitutes 18% of the gross domestic product of US. Furthermore, health spending has increased by 5.4% in 2017 Existence of an aging population, a growing need for care and limited budgets for healthcare force the healthcare clinics to increase efficiency.

Healthcare clinics adopt the strategy of shifting some inpatient services to an outpatient environment in order to decrease the cost and manage the capacity in a more efficient way. This has increased the demand for outpatient care and many new outpatient clinics has been founded to satisfy increasing demand. In this competitive environment, easy access to care, patient satisfaction and short wait times differentiates a health clinic from the others. These factors also contribute to an improvement in patients' health status.

Appointment scheduling system plays the key role in efficiency of healthcare systems. Patients that request an appointment may have different urgency levels and the clinic may give more importance to those patients. In other words, patients may have different priorities. While requesting appointments, patients usually state their preferences about the appointment day, time slot or the physician. There is no doubt that considering patient preferences increases their satisfaction. Patients who are assigned to her preferred appointment day are less likely to cancel their appointment and no-show rates would be lower for those patients (Bowser, Utz, Glick and Harmon, 2010).

This paper constructs a constrained Markov Decision Process (MDP) model for a healthcare facility which considers the revenues and preferences of the patients about the appointment day. In this system, each patient states only one preference. The facility either assigns the patient to her preferred day or rejects the patient. We assume that patients will always show up on their appointment day. To the best of our knowledge, this paper is a new contribution to the literature that models an advanced scheduling problem with patient preferences using constrained MDP. We also derive results about the structure of the optimal policy.

The rest is organized as follows: Section 2 gives a detailed literature review about appointment scheduling. We introduce our problem and describe the model in Section 3. In Section 4, we derive the structure of the optimal policy and introduce solution methodologies. In Section 5, we compare our solution methods with other well-known policies numerically. Finally, we discuss the possible extensions that can be incorporated to our model in Section 6.

## 2. Literature Review

Our model falls into the field of interday advance scheduling problems that consider the preferences of patients with different priorities. In advance scheduling appointment systems, patients are given an appointment for a future day and they are informed about their appointment days beforehand. This section analyzes the existing literature on this area.

We first focus on the literature about within-day appointment scheduling problems. Gupta and Wang (2008) solve a revenue management model for regular and walk-in patients. Regular patients arrive with a specific appointment slot and physician request. The clinic either schedules the patient to his/her preferred slot or rejects the request to preserve slots for walk-in patients. Unlike the model of Gupta and Wang (2008), Wang and Gupta (2011) assume that patients do not have a request for only one specific slot and physician but they have an acceptable set of appointment slots and physicians. They inform the clinic about their acceptable set. The clinic either assigns patients to one of the slots in their acceptable set or rejects them to preserve slots. Thus, patients' acceptable set may include multiple appointment slot-physician pairs whereas in the model of Gupta and Wang (2008), patients' acceptable set consists of only one element. In this manner, this model can be regarded as a generalization of the model of Gupta and Wang (2008). Wang and Fung (2015) consider a totally different appointment scheduling problem where the clinic offers a set of appointment slots and physicians to patients while the previous two papers assume that patients offer their acceptable set to the clinic. In this model, patients either select an option from the offered set or reject the offer. All of these papers consider the intraday scheduling process of a clinic whereas our paper focuses on interday scheduling.

This paragraph explores the papers in interday advance scheduling. Patrick, Puterman and Queyranne (2008) consider the appointment scheduling process of a diagnostic resource. They assume that patients may belong to different priorities and they incorporate the wait-time targets of those patients. Their objective function includes the waiting cost, diversion cost and the cost for neither diverting nor scheduling the patient. Saure, Patrick, Tyldesley and Puterman (2012) extend the model of Patrick et al. (2008) by incorporating multi appointments. Their model is a representation of appointment scheduling in radiation therapy. Gocgun and Puterman (2014) also consider scheduling the chemotherapy patients that require multiple appointments. Patients have target dates and tolerance limits as a must of their treatment. Scheduling a patient on an earlier date or on a later date than the tolerance limit incurs a penalty cost. Truong (2015) studies a multiple-resource appointment scheduling problem with random demand and capacity. He derives analytical results for an advanced appointment scheduling problem for the first time. Parizi and Ghate (2016) consider appointment scheduling problem with multiple resources and multiple patient classes. They take cancellation and no-show behavior of patients into account. The healthcare facility may adopt overbooking strategy to compensate no shows and cancellations. We note that none of the advance scheduling papers up to now consider the patient preferences.

Feldman, Liu, Topaloglu and Ziya (2014) formulate an advance appointment scheduling system for a single resource and single patient type by incorporating patient preferences. They assume that the clinic offers each a patient a set consisting of appointment days. Patients either select a day from the set or reject the offer. They assume that patients' no-show and cancellation probability depends on the time between the call date and the appointment date. They use dynamic programming with the aim of maximizing the expected daily profit. Our model differs from the model of Feldman et al. (2014) in three aspects: Firstly, multiple patient types with different revenues use the clinic in our model. Secondly, instead of offering a portfolio of appointment days to the patients, each patient comes up with an appointment request for a specific day over the booking horizon. The clinic either allocates that specific day for the patient or rejects the patient. Lastly, our model is an infinite horizon discounted constrained MDP model in which we maximize the expected discounted revenue subject to the constraint that expected discounted rejection cost is lower than a predetermined threshold.

Table 1
Comparison Of Recent Papers In Appointment Scheduling

| Properties | Patrick et al. (2008) | Saure et al. (2012) | Feldman et al. (2014) | Gocgun and Puterman (2014) | Truong (2015) | Parizi and Ghate (2016) | Our model |
|---|---|---|---|---|---|---|---|
| Priority&Revenue | + | + | - | + | - | - | + |
| Overbooking | - | - | + | - | - | + | - |
| Multiple resource | - | - | - | - | + | + | - |
| Multiple patient classes | + | + | - | + | + | + | + |
| Random service time/random resource usage | - | - | - | - | + | - | - |
| Cancellations and no-shows | - | - | + | - | - | + | - |
| Deciding on assigning a patient to her day or time preference or not | - | - | - | - | - | - | + |
| Multiple appointments | - | + | - | - | - | - | - |
| Waiting list | + | + | - | + | - | - | - |
| Deciding on the optimal set of appointment days | - | - | + | - | - | - | - |
| Constraint in MDP | - | - | - | - | - | - | + |

To the best of our knowledge, the only paper that included patient preferences over multiple days while making appointment decision is that of Feldman et al. (2014). Ahmadi-Javid, Jalali and Klassen (2017) also emphasize this fact and the limited work in covering patient preferences for appointment days. Table 1 compares the most related papers in interday advance scheduling literature and our model. Moreover, in appointment scheduling literature, there is no paper that formulates an advanced scheduling problem using a constrained MDP. Magerlein and Martin (1978), Cayirli and Veral (2003), Gupta and Denton (2008) and Ahmadi-Javid et al. (2017) provide comprehensive reviews about appointment scheduling.

## 3. Problem Description and Basic Model

This paper considers a healthcare facility resource. There are multiple types of patients with different priorities who want to make an appointment for this resource. “Priority” and “type” are used interchangeably throughout the paper. The priority levels of these patients are based on the potential revenue they will bring to the hospital in our paper. The revenue is obtained after the patient receives the service. We assume that patients do not cancel their appointments and they always show up. Appointment requests for the resource are collected throughout the day and the decision makers make their decision about the appointments at the beginning of the following day before the operations begin. Therefore, the decision epochs correspond to the beginning of a day. The decision makers know the number of newly arriving appointment requests and the number of scheduled appointments for each day over an $N$-day booking horizon on a rolling basis, which will be crucial in constructing the model. Patients can prefer only one day for appointment. Each patient is either given an appointment for his/her preferred day or the patient’s request is rejected with two possible reasons: (1) There is no available appointment slot on the preferred day. (2) The system wants to protect appointment slots on the day she/he prefers for higher-priority patients. In this part, we will introduce the notation of our model. The resource of the facility has the capacity to perform $C$ operations each day. The set of priority classes and the set of appointment days over an $N$-day booking horizon can be expressed as $\mathcal{I} = \{1, 2, \dots, I\}$ and $\mathcal{N} = \{0,1, \dots, N\}$. The revenue obtained from each priority-$i$ patient is $r_i$ and the penalty cost of rejecting a priority-$i$ patient is $\pi_i$. Lower $i$ represents higher priority patients, so $r_i$ decreases as $i$ increases. The facility observes the newly arriving appointment requests from each patient type at the beginning of each day. The maximum number of type- $i$ appointment requests observed at a decision epoch from is $Y_i$. Furthermore, the number of type-$i$ appointment requests observed at a decision epoch is a random variable and we denote it as $W_i$. The random variables $W_i$’s are independent over $i$. In this manner, $W_i$ takes values between 0 and $Y_i$. In our numerical experiments, we assume that it has a truncated Poisson distribution.

When $W_i$ is equal to $w_i$, which occurs with probability $P(W_i = w_i)$, each of these $w_i$ requests may prefer any of the days in the booking horizon. In fact, each priority-$i$ patient prefers $n^{th}$ day from today, which will be called day $n$, with probability $p_{in}$. We note that day preferences of each patient type are independent. In our notation, $R_{in}$ represents the random variable for the number of newly arriving appointment requests from priority $i$ patients for day $n$. In the rest of the paper, we focus on the random vector $\vec{R}_i = (R_{i0}, R_{i1}, \dots, R_{in}, \dots, R_{iN})$ since $R_{i0}, R_{i1}, \dots, R_{iN}$ have joint probability distribution for a given $i$ value. Given the event that $W_i = w_i$, $\vec{R}_i$ is equal to $u_i$ with probability $P(\vec{R}_i = \vec{u}_i | W_i = w_i)$, where $\vec{u}_i = (u_{i0}, u_{i1}, \dots, u_{in}, \dots, u_{iN})$ and $w_i = \sum_n u_{in}$. This means that $w_i$ appointment requests are partitioned over $N$ days in the booking horizon. Moreover, the distribution of $\vec{R}_i | W_i$ corresponds with the multinomial distribution:

$$P(\vec{R}_i = \vec{u}_i | W_i = w_i) = \frac{w_i!}{u_{i0}! u_{i1}! \dots u_{iN}!} p_{i0}^{u_{i0}} p_{i1}^{u_{i1}} \dots p_{iN}^{u_{iN}} \quad (1)$$

The clinic observes the realization of $R_{in}$ at each decision epoch. Throughout the paper, $u_{in}$ and $y_{in}$ are used to represent the realizations of $R_{in}$. The summary of the notation used in the model can be found in Table 2.

Table 2
Notation Summary For the Model

| Notation | Description |
|---|---|
| $\mathcal{I} = \{1,2,\dots,I\}$ | Set of priority classes |
| $\mathcal{N} = \{0,1,\dots,N\}$ | Set of appointment days over an $N$-day booking horizon |
| $Y_i$ | Maximum number of appointment requests observed at a decision epoch from type- $i$ patients |
| $W_i$ | Random variable that represents the number of type- $i$ appointment requests observed at a decision epoch |
| $p_{in}$ | Probability that a priority i patient prefers $n^{th}$ day from now |
| $R_{in}$ | Random variable that represents the number of priority-$i$ patients who prefer $n^{th}$ day from now |
| $P(\vec{R}_i = \vec{u}_i \mid W_i = w_i)$ | Given that the number of type- $i$ appointment requests observed at a decision epoch is $w_i$, the joint probability that the number of type-$i$ patients who prefer day $n$ is $u_{in}$ for $n \in \mathcal{N}$ |
| $C$ | The number of available appointment slots each day |
| $r_i$ | Revenue obtained from a priority-$i$ patient |
| $\pi_i$ | Penalty cost of rejecting a priority-$i$ patient |

### 3.1 The State Space

The state of the system can be written in the following way:

$$\vec{s} = (\vec{u},\vec{x}) = (u_{10}, u_{20}, \dots, u_{I0}, u_{11}, u_{21}, \dots, u_{I1}, \dots, u_{in}, \dots, u_{1N}, u_{2N}, \dots, u_{IN};$$
$$x_{10}, x_{20}, \dots, x_{I0}, x_{11}, x_{21}, \dots, x_{I1}, \dots, x_{in}, \dots, x_{1(N-1)}, x_{2(N-1)}, \dots, x_{I(N-1)}) \qquad (2)$$

where $u_{in}$ is the number of newly arriving appointment requests from priority-$i$ patients for day $n$ and $x_{in}$ is the number of appointments for type-$i$ patients already booked on day $n$. In this manner, $x_{i0}$ represents the number of type-$i$ appointments booked for today.

$$S = \left\{ \begin{matrix} (\vec{u},\vec{x}) \in (\mathbb{Z}_I^+ \times \mathbb{Z}_{N+1}^+) \times (\mathbb{Z}_I^+ \times \mathbb{Z}_N^+) \mid \sum_{i\in\mathcal{I}} x_{in} \le C,\ 0 \le n \le N-1; \\ \sum_{n\in\mathcal{N}} u_{in} \le Y_i\, , 1 \le i \le I \end{matrix} \right\} \qquad (3)$$

### 3.2 The Action Set

At each decision epoch, the set of actions is the number of priority i patients booked on day $n$, $a_{in}$. The set of feasible actions at state $\vec{s}$ is

$$A_{\vec{s}} = \left\{ \begin{matrix} \vec{a} \in \mathbb{Z}_I^+ \times \mathbb{Z}_{N+1}^+ \mid \sum_{i\in\mathcal{I}} (a_{in} + x_{in}) \le C,\ 0 \le n \le N-1; \sum_{i\in\mathcal{I}} a_{iN} \le C; \\ a_{in} \le u_{in}, 0 \le n \le N, i \le i \le I \end{matrix} \right\} \qquad (4)$$

### 3.3 State Transition Probabilities and Preference Probabilities

After the decision is made, new appointment requests occur throughout the day and that is the only random event in our model. As we defined earlier, the state of the system at the beginning of the day before the decision can be represented as below:

$$\vec{s} = (\vec{u}, \vec{x}) = (u_{10}, u_{20}, \dots, u_{I0}, u_{11}, u_{21}, \dots, u_{I1}, \dots, u_{in}, \dots, u_{1N}, u_{2N}, \dots, u_{IN}; \\ x_{10}, x_{20}, \dots, x_{I0}, x_{11}, x_{21}, \dots, x_{I1}, \dots, x_{in}, \dots, x_{1(N-1)}, x_{2(N-1)}, \dots, x_{I(N-1)}) \qquad (5)$$

By taking the rolling time horizon into account, the next state of the system at the beginning of the next decision epoch is

$$\vec{s}' = (\vec{y}, \vec{x}') = (y_{10}, y_{20}, \dots, y_{I0}, y_{11}, y_{21}, \dots, y_{I1}, \dots, y_{in}, \dots, y_{1N}, y_{2N}, \dots, y_{IN}; \\ x_{11} + a_{11}, x_{21} + a_{21}, \dots, x_{I1} + a_{I1}, \dots, x_{in} + a_{in}, \dots, x_{1(N-1)} + a_{1(N-1)}, x_{2(N-1)} + a_{2(N-1)}, \dots, \\ x_{I(N-1)} + a_{I(N-1)}, a_{1N}, a_{2N}, \dots, a_{iN}, \dots, a_{IN}) \qquad (6)$$

The transition from state $\vec{s}$ to $\vec{s}^{'}$ occurs with the following probability:

$$P(\vec{s}^{'}|\vec{s}, \vec{a}) = \prod_{i\in\mathcal{I}} P(\vec{R}_i = \vec{y}_i) = \prod_{i\in\mathcal{I}} P(W_i = w_i) P(\vec{R}_i = \vec{y}_i | W_i = w_i) \qquad (7)$$

In Equation (7), we remind that $P(\vec{R}_i = \vec{y}_i | W_i = w_i)$ is given in Equation (1), $w_i = \sum_n y_{in}$ and $\vec{R}_i = (R_{i0}, R_{i1}, \dots, R_{in}, \dots, R_{iN})$ and $\vec{y}_i = (y_{i0}, y_{i1}, \dots, y_{in}, \dots, y_{iN})$. The multiplication follows from the independence of $W_i$'s and day preferences over $i$. The transition from $\vec{x}$ to $\vec{x}$ is not random since it is a function of the action $\vec{a}$. The stochasticity emerges in the number of newly observed requests at the beginning of the next day.

### 3.4 Cost Criteria

We maximize the infinite horizon expected discounted revenue subject to the constraint that expected discounted rejection cost does not exceed a user-specified constant. Given the initial state $\vec{s} \in S$, expected discounted revenue under an arbitrary policy $\psi$ is

$$K_\psi(\vec{s}) = E(\textstyle\sum_{t=0}^{\infty} \beta^t H(\vec{s}_t, \vec{a}_t) \,|\vec{s}_0 = \vec{s}) \qquad (8)$$

where $\vec{s}_t$ is the state at time $t$ under policy $\psi$, $H(\vec{s}_t, \vec{a}_t) = H\big((\vec{u}_t, \vec{x}_t\,), \vec{a}_t\big) = \sum_{i\in\mathcal{I}} r_i(x_{i0} + a_{i0})$ is the immediate reward and $\beta$ is the discount factor. Similarly, given the initial state $\vec{s} \in S$, expected discounted rejection cost under an arbitrary policy $\psi$ is

$$C_\psi(\vec{s}) = E(\textstyle\sum_{t=0}^{\infty} \beta^t D(\vec{s}_t, \vec{a}_t) \,|\vec{s}_0 = \vec{s}) \qquad (9)$$

where

$$D(\vec{s}_t, \vec{a}_t) = D\big((\vec{u}_t, \vec{x}_t\,), \vec{a}_t\big) = \textstyle\sum_{i\in\mathcal{I}, n\in\mathcal{N}} \pi_i(u_{in} + a_{in})$$

is the immediate penalty for rejecting patients.

Therefore, our problem can be expressed in the following way:

$$\sup_{\psi\in\mathbb{A}} K_\psi(\vec{s}) \\ s.t.\ C_\psi(\vec{s}) \le c \qquad (10)$$

where $\mathbb{A} = \prod_{\vec{s}\in S} A_{\vec{s}}$ is the set of all feasible actions.

## 4. Solution Methodologies and the Structure of the Optimal Policy

### 4.1. Relaxed Version of the Constrained Problem and Bellman Equations

In order to analyze the structure of the optimal policy, we first introduce the Lagrangian multipliers and obtain an unconstrained MDP by relaxing the constraint with the Lagrangian multiplier $\theta \geq 0$. Our problem becomes

$$V^{\theta}(\vec{s}) = \sup_{\psi \in \mathbb{A}} V^{\theta}_{\psi}(\vec{s}) \tag{11}$$

where

$$V^{\theta}(\vec{s}) = E(\textstyle\sum_{t=0}^{\infty} \beta^{t}(H(\vec{s}_t, \vec{a}_t) - \theta D(\vec{s}_t, \vec{a}_t))\,|\vec{s}_0 = \vec{s}) = K_{\psi}(\vec{s}) - \theta C_{\psi}(\vec{s}) \tag{12}$$

From now on, we call problem (11) and (12) "relaxed problem." The value function $V^{\theta}(\vec{s})$ represents the $\theta$-optimal infinite horizon objective function of the relaxed problem given the initial state of the system is $\vec{s} \in S$. Moreover, $\vec{s}'$ is determined using the transition in Equation (6). Given that $\beta$ is the discount factor, the value function of our model can be written for every $\vec{s} \in S$ in the following way:

$$V^{\theta}(\vec{s}) = \max_{\vec{a} \in A_{\vec{s}}} \textstyle\sum_{i \in \mathcal{I}} r_i(x_{i0} + a_{i0}) - \theta \sum_{i \in \mathcal{I}, n \in \mathcal{N}} \pi_i(u_{in} - a_{in}) + \beta \sum_{\vec{y} \in \mathbb{Y}} P(\vec{y})\, V^{\theta}(\vec{s}') \tag{13}$$

where $\mathbb{Y} = \{\vec{y} \in \mathbb{Z}_I^{+} \times \mathbb{Z}_{N+1}^{+} | \sum_n y_{in} \leq Y_i, 1 \leq i \leq I\}$.

Any stationary policy that maximizes Equation (13) is called $\theta$-optimal policy. Assuming that there exists a policy $\psi$ satisfying $K_{\psi}(\vec{s}) + C_{\psi}(\vec{s}) < \infty$ for all $\vec{s} \in S$, $\theta$-optimal value function, $V^{\theta}(\vec{s})$, is finite. We show in the appendix that this is a valid assumption for our problem. However, finding the optimal solution of (13) is impossible for large instances since the exact solution is intractable due to high-dimensional state space.

Puterman (1994) shows that (13) is equivalent to the following LP formulation:

$$\min_{\overrightarrow{V^{\theta}}} \sum_{(\vec{u}, \vec{x}) \in S} \alpha(\vec{u}, \vec{x}) V^{\theta}(\vec{u}, \vec{x})$$

*subject to*

$$\max_{\vec{a} \in A_{\vec{s}}} \textstyle\sum_{i \in \mathcal{I}} r_i(x_{i0} + a_{i0}) - \theta \sum_{\substack{i \in \mathcal{I}, \\ n \in \mathcal{N}}} \pi_i(u_{in} - a_{in}) + \beta \sum_{\vec{y} \in \mathbb{Y}} P(\vec{y})\, V^{\theta}(\vec{y}, \vec{x}') \leq V^{\theta}(\vec{u}, \vec{x}) \tag{14}$$

$$\forall \vec{a} \in A_{(\vec{u}, \vec{x})} \; and \; (\vec{u}, \vec{x}) \in S$$

where $\alpha(\vec{u}, \vec{x})$ is the probability that initial state of the system is $(\vec{u}, \vec{x})$. Furthermore, $(\vec{u}, \vec{x})$ and $(\vec{y}, \vec{x}')$ are defined in Equation (5) and Equation (6), respectively.

It is obvious that problem (14) also suffers from the curse of dimensionality for large instances since it has as many variables as the number of possible state-action pairs.

In the following sections, the solution methodologies in reinforcement learning will be discussed to obtain a good solution to (13).

### 4.2 Structure of the Optimal Policy

In order to prove the structure of the optimal policy, we need to make the following assumptions given by Sennott (1991):

*Assumption 1:* There exists a policy $\psi$ satisfying $K_{\psi}(\vec{s}) + C_{\psi}(\vec{s}) < \infty$ for all $\vec{s} \in S$.

*Assumption 2:* There exists a policy $\Phi$ satisfying $K_{\Phi}(\vec{s}) < \infty$ and $C_{\Phi}(\vec{s}) < c$ given that the initial state is $\vec{s}$.

The detailed analysis about the assumptions can be found in Appendix. Below, we introduce the definitions of "booking limit policy" and "mixed policy"'.

*Definition 1:* Let $\psi$ be a policy implemented in the following way:

Suppose that $\vec{b} = (\vec{b}_1, \vec{b}_2, \dots, \vec{b}_n, \dots, \vec{b}_{N+1})$ is the vector of thresholds, where each $\vec{b}_n$ is an $I$ dimensional vector such that $\vec{b}_n = (b_{1n}, b_{2n}, \dots, b_{in}, \dots, b_{In})$. Here, $b_{in}$ is the strict booking limit implemented on day $n$ for type-$i$ patients. Furthermore, suppose that $x_{in}$ is the number of type-$i$ patients already booked on day $n$. A type-$i$ patient preferring day $n$ accepted if $\sum_{l=1}^{I} x_{ln} < b_{in}$.

Then, $\psi$ is a strict booking limit policy with parameter $\vec{b}$.

*Definition 2:* In this paper, a mixed policy $(p, f, e)$ can be defined as a randomized stationary policy that chooses policy $f$ with probability $p$ and policy $e$ with probability $1 - p$ at each stage for any $0 \leq p \leq 1$, where $f$ and $e$ are stationary polices.

**Proposition 1:** Assume that Assumption 1 holds. The value function $V^{\theta}(\vec{s})$ satisfying the right-hand side of Equation (13) is concave in $\vec{x}$. This means that the following inequality holds for every $(\vec{u},\vec{x}) \in S$ and every combination of $i$ and $n$:

$$V_t^\theta(\vec{u},\vec{x}+2e_{in}) - 2V_t^\theta(\vec{u},\vec{x}+e_{in}) + V_t^\theta(\vec{u},\vec{x}) \le 0$$

where $e_{in}$ is a matrix consisting of 1 in the $(i,n)^{th}$ position and zeros elsewhere. As a result, the optimal policy is the strict booking limit policy for the problem (11).

It is important to note that the optimal booking limit policy has different booking limits for each day in the booking horizon.

Application of the optimal policy: Let $\zeta_{in} = \sum_{\vec{y}\in\mathbb{Y}} P(\overrightarrow{y})\, \Delta V\,(\vec{y},\vec{x}+e_{in}\,) + \theta\frac{\pi_i}{\beta}$. While applying the optimal policy, we first need to order $(i,n)$ pairs from the highest to the lowest according to their $\zeta_{in}$ values. It is required to start with the $(i,n)$ in the first rank, which represents type-$i$ patients who prefer day $n$ and continue until the total number of accepted patients exceeds the capacity $C$ on day $n$. Then, we need to continue with $(i,n)$ in the second rank and continue until $\sum_{\vec{y}\in\mathbb{Y}} P(\overrightarrow{y})\, \Delta V\,(\vec{y},\vec{x}+e_{in}\,) < -\theta\frac{\pi_i}{\beta}$ or the total number of accepted patients exceeds the capacity $C$ on day $n$. If $n = 0$, there is no need to protect any slots for the subsequent days. Therefore, it is required to stop accepting the patients to day 0 only if the number of occupied slots reach $C$. The rest continues similarly. This procedure implies that there are booking limits $b_{in}$ such that one should accept a type-$i$ patient preferring day $n$ only if the number of occupied appointment slots is less than a prespecified threshold, $\sum_{l=1}^{I} x_{ln} \le b_{in}$.

**Theorem 1:** Assume that Assumptions 1 and 2 hold. The optimal policy for problem (10) is the mixture of at most two booking limit policies. Furthermore, there exists at most one state for which those two stationary policies differ.

The proofs of the Proposition 1 and Theorem 1 are given in Appendix. The proof mainly relies on the paper of Sennott (1991).

Example: We conducted a numerical experiment for a small-instance. The parameters were set in the following way:

$$c = 419, N = 2, I = 2, C = 2, r_1 = 200, r_2 = 100, \pi_1 = 37.5\ and\ \pi_2 = 25$$

Furthermore, the number of arriving type-1 and type-2 patients has Poisson distribution with means 1 and 0.5, respectively. To protect the finiteness of the state space, we truncated the Poisson distribution such that the maximum number of type-$i$ patients requesting an appointment is twice the mean number of arrivals for type-$i$ patients. We obtained the optimal solution by constructing the LP representation for the constrained MDP model. Note that for day 0, it can be easily seen that the following action is optimal: (1) Schedule type-1 patients who prefer day 0 until there are no available slots. (2) If there are still available slots, schedule type-2-patients preferring day 0 until all of the slots become occupied. For stationary policy 1, the booking limit vector for day 1 was $\vec{b}_1^1 = [2,2]$ and the booking limit vector for day 2 was $\vec{b}_2^1 = [2,1]$. For stationary policy 2, the booking limit vector for day 1 was $\vec{b}_1^2 = [2,2]$ and the booking limit vector for day 2 was $\vec{b}_2^2 = [2,2]$. The randomization probability was 0.0586. According to the optimal policy, the state for which those two stationary policies differ was $\vec{s}^* = (\vec{u}^*,\vec{x}^*)$, where $\vec{u}^* = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ and $\vec{x}^* = \begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix}$. This means that the optimal action for the states other than $\vec{s}^*$ will be determined using either $\vec{b}^1$ or $\vec{b}^2$. At state $\vec{s}^*$, one should apply stationary policy 1 with the probability of 0.0586 and apply stationary policy 2 with the probability of 0.9414.

After that point, we implemented the methods given in the following sections to obtain good booking limits.

### 4.3 Approximate Linear Programming

We mainly use approximate LP in order to provide a good initial value functions to TD Learning. The performance comparison of this method with the others is given in the numerical experiments. In order to deal with the high-dimensional state, one of the methods is value function approximation, in which the value function is approximated with a parametric function. The optimal policy is determined given that the value function lies in the set of functions with that specific parametric form. In order to have an idea about the structure of the value function and start with a good approximation, we have calculated the exact solution for small instances by setting $C$ to 4, $I$ to 2 and $N$ to 3. We conducted regression where different features of the state are independent variables and the value function is the dependent variable. As a result, we decided on the following approximation:

$$V^\theta(\vec{s}) \approx w_0 + \sum_{i=1}^{I} w_i \sum_{n=0}^{N} u_{in} + \sum_{i=1}^{I} \sum_{n=0}^{N-1} v_{in} x_{in} \qquad (15)$$

R-square value of the regression was 91.42% when $\sum_{n=0}^{N} u_{in}$ and $x_{in}$ were taken as independent variables. When the interaction terms and second-order terms were included, R-square increased to 93.75%. Since the difference was small, the linear approximation was used for the sake of simplicity. When the approximation in (15) was plugged to LP model in (14), we obtain the primal LP below:

$$\min_{\vec{v},\vec{w}} w_0 + \sum_{i=1}^{I} w_i \sum_{n=0}^{N} E_\alpha[u_{in}] + \sum_{i=1}^{I}\sum_{n=0}^{N-1} v_{in} E_\alpha[x_{in}]$$

$$s.t.\ (1-\beta)w_0 + \sum_{i=1}^{I}\sum_{n=0}^{N-2} v_{in}\left(x_{in} - \beta\left(x_{i(n+1)} + a_{i(n+1)}\right)\right) + \sum_{i=1}^{I}\left(x_{i(N-1)} - \beta a_{iN}\right)$$
$$+ \sum_{i=1}^{I}\sum_{n=0}^{N} w_i\left(u_{in} - \beta E[y_{in}]\right) \geq \sum_{i=1}^{I} r_i(x_{i0} + a_{i0}) - \theta \sum_{i=1}^{I}\sum_{n=0}^{N} \pi_i(u_{in} - a_{in})$$

$$\forall \vec{a} \in A_{(\vec{u},\vec{x})}\ and\ (\vec{u},\vec{x}) \in S$$

$$\vec{v},\vec{w} \geq 0 \qquad (16)$$

With the approximated value function, we could decrease the number of variables to $I + IN + 1$. However, since there exists a constraint for every state-action pair, there is still curse of dimensionality problem in the LP formulation (16) due to the large number of constraints. Given dual variable $\vec{X}$, the dual formulation of (16) is

$$\max_{\vec{X}} \textstyle\sum_{(\vec{u},\vec{x})\in S} \sum_{\vec{a}\in A_{(\vec{u},\vec{x})}} \vec{X}((\vec{u},\vec{x}),\vec{a}) \left(\sum_{i=1}^{I} r_i(x_{i0} + a_{i0}) - \theta \sum_{i=1}^{I}\sum_{n=0}^{N} \pi_i(u_{in} - a_{in})\right)$$

$$s.t.\ (1-\beta) \textstyle\sum_{(\vec{u},\vec{x})\in S} \sum_{\vec{a}\in A_{(\vec{u},\vec{x})}} \vec{X}\left((\vec{u},\vec{x}),\vec{a}\right) = 1$$

$$\textstyle\sum_{(\vec{u},\vec{x})\in S} \sum_{\vec{a}\in A_{(\vec{u},\vec{x})}} \vec{X}((\vec{u},\vec{x}),\vec{a}) \left(x_{in} - \beta\left(x_{i(n+1)} + a_{i(n+1)}\right)\right) \leq E_\alpha[x_{in}]$$

$$\forall n = 0,1,\dots,N-2\ and\ \forall i = 1,2,\dots,I$$

$$\textstyle\sum_{(\vec{u},\vec{x})\in S} \sum_{\vec{a}\in A_{(\vec{u},\vec{x})}} \vec{X}((\vec{u},\vec{x}),\vec{a}) \left(x_{i(N-1)} - \beta a_{iN}\right) \leq E_\alpha[x_{i(N-1)}] \quad \forall i = 1,2,\dots,I$$

$$\textstyle\sum_{(\vec{u},\vec{x})\in S} \sum_{\vec{a}\in A_{(\vec{u},\vec{x})}} \vec{X}((\vec{u},\vec{x}),\vec{a}) \sum_{n=0}^{N} w_i(u_{in} - \beta E[y_{in}]) \leq \sum_{n=0}^{N} E_\alpha[u_{in}] \quad \forall i = 1,2,\dots,I$$

$$\vec{X} \geq 0 \qquad (17)$$

With the dual formulation, although the number of constraints decreases drastically, there are as many variables as the number of possible state-action pairs. Luckily, we can benefit from the column generation algorithm to solve LP model given in (16) and (17). When it is solved, the optimal $\vec{v}$, $\vec{w}$ and $w_0$ are obtained for the approximate LP model. While comparing the performance of approximate LP in the simulation, it is necessary to obtain the approximate optimal policy. To achieve this, we plug $w_0 + \sum_{i=1}^{I} w_i \sum_{n=0}^{N} y_{in} + \sum_{i=1}^{I} \sum_{n=0}^{N-1} v_{in} x'_{in}$ for $V^{\theta}(\vec{s}')$ into Equation (13). As a result, one needs to solve the following MIP to obtain the approximate optimal policy at each decision epoch:

$$\max_{\vec{a} \in A_{\vec{s}}} \{ \sum_{i \in \mathcal{I}} r_i (x_{i0} + a_{i0}) - \theta \sum_{i \in \mathcal{I}, n \in \mathcal{N}} \pi_i (u_{in} - a_{in}) + \beta \sum_{\vec{y} \in \mathbb{Y}} P(\vec{y}) \big( w_0 + \sum_{i=1}^{I} \sum_{n=0}^{N-2} v_{in} \big( x_{i(n+1)} + a_{i(n+1)} \big) + \\ \sum_{i=1}^{I} v_{iN} a_{iN} + \sum_{i=1}^{I} w_i \sum_{n=0}^{N} y_{in} \big) \} \tag{18}$$

Patrick et al. (2008) follow the same method as the one described in this section to solve their advance scheduling problem.

### 4.4 TD Learning

TD learning is one of the central algorithms of reinforcement learning. Using TD methods, one can directly learn from the experience without assuming a model family for the value function. In this method, the value function is updated at each iteration using the temporal difference, which is the difference between the estimated value of being in a particular state and the actual value. If one wants the weight of the recent updates to be higher than the earlier updates, an artificial discount factor, $\lambda$, is introduced, where $0 \leq \lambda \leq 1$. In addition to estimating the value function of a given policy, TD learning is also used to determine an approximately optimal policy. In TD learning algorithms designed for infinite horizon problems, while determining the next state to visit using value function approximation, we also update the value of being in the states visited up to then. Further information about TD learning can be found in Powell (2011), Sutton and Barto (2011) and Sugiyama (2015).

Using TD learning, we tried to estimate the value function of a given booking limit policy. Algorithm 1 gives the steps followed to determine the value of the booking limit policy $\phi$. This algorithm was one of the algorithms given by Powell (2011) with a slight modification.

Algorithm 1

TD($\lambda$) Algorithm

```
Step 0.    Initialization
               Step 0a.   Initialize $V_0^{\theta}(\vec{u}, \vec{x})$ for all $(\vec{u}, \vec{x}) \in S$.
               Step 0b.   Initialize the state $(\vec{u}_0, \vec{x}_0)$.
               Step 0c.   Set $t = 1$.
Step 1.    Generate a sample for $\vec{y}_t$.
Step 2.    Compute the temporal difference for this step:
               $\gamma^{\phi,t} = \sum_{i=1}^{I} r_i (x_{i0}^t + a_{i0}^{\phi,t}) - \theta \sum_{i,n} \pi_i \big( u_{in}^t - a_{in}^{\phi,t} \big)$
                   $+ \beta \big( V_{t-1}(\vec{u}_t, \vec{x}'_t) - V_{t-1}(\vec{u}_t, \vec{x}_t) \big)$
           where $\vec{x}'_t$ is determined using the transition in Equation (6).
```

```
Step 3.   Update $V^{\theta}(.)$ for $m = t, t-1, \dots, 1$:
          $V_t(\vec{u}_m, \vec{x}_m) = V_{t-1}(\vec{u}_m, \vec{x}_m) + (\beta\lambda)^{t-m}\gamma^{\phi,t}$ .
Step 4.   Compute $\vec{s}_{t+1} = \vec{s}'_t = (\vec{y}_{t+1}, \vec{x}'_t)$ where $\vec{y}_{t+1}$ is generated in Step 1 and $\vec{x}'_t$ is
          determined using the transition in Equation (6).
Step 5.   Set $t = t + 1$. If $t < T$, return to Step 1. Otherwise, stop.
```

In this algorithm, we first generate a sample for $\vec{y}$, which is the number of patients who prefer a particular day. In Step 2, we calculate the temporal difference. In Step 3, we update the value function of all visited states using the temporal difference found in Step 2. After calculating the next state in Step 4, we continue with the next iteration if the number of iterations executed does not exceed $T$, which is the maximum number of iterations allowed. We initialize TD learning algorithm using the approximate value function obtained by plugging the optimal $\vec{v}, \vec{w}$ and $w_0$ into Equation (15).

## 5. Numerical Experiments

In this section, we conducted experiments on the relaxed problem given in Equation (11) and (12). First, we tested the performance of approximate LP on a small instance. The experimental setting can be described in the following way: There are two patient types and three days in the booking horizon. The number of type-$i$ patients requesting an appointment has Poisson distribution with mean $g_i$. To protect the finiteness of the state space, we truncated the Poisson distribution such that the maximum number of type-$i$ patients requesting an appointment is twice the mean number of arrivals for type-$i$ patients. The parameter setting was

$$\beta = 0.99, \theta = 1, g_1 = 1.5, g_2 = 0.5, C = 4, r_1 = 200, r_2 = 100, \pi_1 = 37.5, \pi_2 = 25$$

For this small instance, the comparison between the solution of approximated LP and that of the exact DP, found using value iteration, showed that the average difference and maximum difference across all the states were 2.39% and 5.06%, respectively.

Secondly, we tested the convergence performance of TD learning in the following experimental setting: There are three patient types and nine days in the booking horizon. The number of newly arrived type-$i$ patients has a truncated Poisson distribution with mean $g_i$. Parameters were set as

$$\beta = 0.99, \theta = 1, g_1 = 20, g_2 = 15, g_3 = 20, C = 40, r_1 = 200, r_2 = 100, r_3 = 50, \pi_1 = 150, \pi_2 = 100, \pi_3 = 75$$

Given that $b_i$ is the strict booking limit for type-$i$ patients and the same booking limits are applied on each day over the booking horizon, you can see the convergence of the value function for $b_1 = 40$, $b_2 = 16$ and $b_3 = 3$ in Figure 1.

It converged in 200 iterations, which takes approximately 15 seconds.

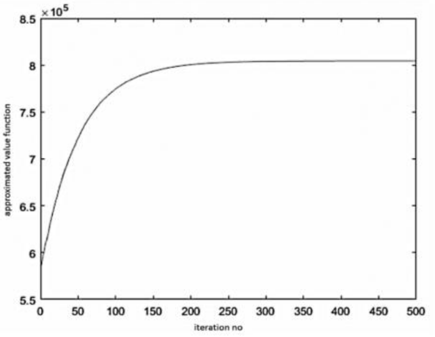

Figure 1. Value Function For The Strict Booking Limit [40 16 3] Through Iterations

Finally, we conducted a simulation study to compare the performance of the booking limit policy calculated using TD learning and the other policies. The simulation setting can be summarized in the following way: There are three patient types and nine days in the booking horizon, the number of newly arrived patient of type-$i$ has Poisson distribution with mean $g_i$. As before, in order to protect the finiteness of the state space, we truncated the Poisson distribution such that the maximum number of type-$i$ patients requesting an appointment is three times the mean number of arrivals for type-$i$ patients. The parameters were set as below:

$$\beta = 0.99, \theta = 1, C = 30, r_1 = 200, r_2 = 100, r_3 = 50,\ \pi_1 = 150, \pi_2 = 100,\ \pi_3 = 75$$

We simulated the system for 6 different scenarios where $g_i$ values are given in the first column of Table 3. The simulation length was 500 days and the warm-up period was 10 days. We collected the statistics for the objective function value of the relaxed problem with 95% confidence interval, average percentage of diverted patients and average utilization of each policy. For the approximate LP, we coded the column generation algorithm in Matlab and used cvx-solver for the integer program in Equation (18). We used TD learning to determine a good simple strict-booking limit policy such that the booking limits are the same for each day over the booking horizon. Such a policy is more preferable since implementing it is easier in practice. At this point, we found the approximate value function of a given simple strict booking limit policy using TD learning. It is also essential to emphasize that the value function was initialized using $w_0$, $w_i$ and $v_{in}$, obtained through approximate LP. After estimating the value functions for each booking limit policy, we selected the one with the highest approximate value function. You can see the results in Table 3. TD booking limit policy represents the strict booking limit calculated with TD learning algorithm. Expected booking limit policy represents the booking limit policy calculated using the expected number of arriving patients for each type. The booking limits calculated for each scenario using these two methods can be seen in the first column of Table 3. $\vec{b}_e$ represents the expected booking limit and $\vec{b}_t$ represents the TD booking limit. Finally, using $w_0$, $w_i$ and $v_{in}$ values obtained by applying the column generation algorithm to approximate LP given in (16) and (17), we calculated the approximate LP policy by solving Problem (18). The results show that TD booking limit policy performs better than all other policies since under this policy, the objective function value of the relaxed problem is higher and the divergence rate is lower for higher-priority patient types.

## 6. Conclusion

This paper focuses on the appointment scheduling mechanism of a physician. Patients from multiple priority classes arrive at the facility and request an appointment by informing the clinic about their most preferred appointment day. The clinic either accepts the request or rejects the patient to protect slots for higher-priority patients. We model such a system using discrete time Markov decision process to maximize the long-run discounted revenue subject to the constraint that long-run discounted rejection cost is below a specific threshold. We prove that the optimal policy of this model is a randomized booking-limit policy. Since the problem cannot be solved optimally due to high-dimensional state, we apply Approximate Dynamic Programming methods. First, we approximate the value function with a linear function and solve the resulting LP using column generation. Then, by initializing the TD learning algorithm with this solution, we approximately calculate the value function arising from implementing each possible booking limit policy. Our numerical results show that TD learning algorithm performs well in determining the value of a booking-limit policy. The booking limit calculated using TD learning gives higher value than other policies.

Table 3
Summary of the Simulation Results

| Scenario | Performance Measure | Patient type | TD Booking Limit Policy | Expected Booking Limit Policy | Approximate LP |
|---|---|---|---|---|---|
| $\vec{g} = [10, 15, 20]$ | Rate of diverted patients | Type-1 | 0.0174∓0.0018 | 0.0168∓0.0018 | 0.1868∓0.0048 |
| | | Type-2 | 0.1172∓0.0026 | 0.1214∓0.0051 | 0.3172∓0.0055 |
| $\vec{b}_t = [30, 21, 9]$ | | Type-3 | 0.6499∓0.0030 | 0.6612∓0.0031 | 0.4194∓0.0020 |
| $\vec{b}_e = [30, 20, 5]$ | Utilization | | 0.9935∓0.0011 | 0.9937∓0.0007 | 0.9995∓0.0002 |
| | Objective func. value | | 309,960∓226.19 | 295,140∓299.77 | 230,320∓261.88 |
| $\vec{g} = [10, 20, 15]$ | Rate of diverted patients | Type-1 | 0.1306∓0.0067 | 0.0105∓0.0021 | 0.1920∓0.0037 |
| | | Type-2 | 0.2959∓0.0052 | 0.1036∓0.0036 | 0.3319∓0.0031 |
| $\vec{b}_t = [30, 27, 8]$ | | Type-3 | 0.5008∓0.0016 | 0.8839∓0.0028 | 0.4277∓0.0037 |
| $\vec{b}_e = [30, 20, 0]$ | Utilization | | 0.9995∓0.0002 | 0.9840∓0.0016 | 0.9994∓0.0002 |
| | Objective func. value | | 325,090∓212.08 | 306,800∓278.49 | 238,660∓216.78 |
| $\vec{g} = [15, 10, 20]$ | Rate of diverted patients | Type-1 | 0.1071∓0.0036 | 0.0313∓0.0034 | 0.2185∓0.0058 |
| | | Type-2 | 0.2620∓0.0032 | 0.1498∓0.0061 | 0.3375∓0.0052 |
| $\vec{b}_t = [30, 27, 9]$ | | Type-3 | 0.5275∓0.0031 | 0.6606∓0.0041 | 0.4194∓0.0031 |
| $\vec{b}_e = [30, 15, 5]$ | Utilization | | 0.9990∓0.0004 | 0.9943∓0.0011 | 0.9994∓0.0002 |
| | Objective func. value | | 358,230∓259.73 | 344,170∓321.02 | 261,440∓297.13 |
| $\vec{g} = [15, 20, 10]$ | Rate of diverted patients | Type-1 | 0.2083∓0.0027 | 0.0436∓0.0023 | 0.2173∓0.0072 |
| | | Type-2 | 0.3690∓0.0043 | 0.2449∓0.0063 | 0.3654∓0.0057 |
| $\vec{b}_t = [30, 20, 14]$ | | Type-3 | 0.4229∓0.0045 | 0.9581∓0.0036 | 0.4391∓0.0080 |
| $\vec{b}_e = [30, 15, 0]$ | Utilization | | 0.9995∓0.0002 | 0.9944∓0.0009 | 0.9996∓0.0002 |
| | Objective func. value | | 359,140∓222.56 | 344,470∓307.46 | 266,080∓386.48 |
| $\vec{g} = [20, 10, 15]$ | Rate of diverted patients | Type-1 | 0.2145∓0.0034 | 0.0310∓0.0027 | 0.2426∓0.0051 |
| | | Type-2 | 0.4086∓0.0046 | 0.1547∓0.0063 | 0.3697∓0.0056 |
| $\vec{b}_t = [30, 14, 12]$ | | Type-3 | 0.4172∓0.0023 | 0.8840∓0.0029 | 0.4284∓0.0043 |
| $\vec{b}_e = [30, 10, 0]$ | Utilization | | 0.9993∓0.0003 | 0.9843∓0.0022 | 0.9995∓0.0002 |
| | Objective func. value | | 409,060∓270.07 | 392,510∓396.44 | 286,420∓317.28 |
| $\vec{g} = [20, 15, 10]$ | Rate of diverted patients | Type-1 | 0.1787∓0.0030 | 0.0595∓0.0032 | 0.2411∓0.0059 |
| | | Type-2 | 0.3434∓0.0062 | 0.2908∓0.0059 | 0.3823∓0.0029 |
| $\vec{b}_t = [30, 20, 4]$ | | Type-3 | 0.6024∓0.0023 | 0.9522∓0.0036 | 0.4365∓0.0034 |
| $\vec{b}_e = [30, 10, 0]$ | Utilization | | 0.9989∓0.0003 | 0.9933∓0.0010 | 0.9994∓0.0003 |
| | Objective func. value | | 412,850∓225.88 | 391,790∓366.77 | 299,120∓376.53 |

While formulating this problem, we use some simplifications. First, it is assumed that patients have only one preference. Nevertheless, patients can be given a flexibility to inform the clinic about their acceptable set of appointment days and the clinic can assign the patients to one of the days in their acceptable set or reject them. The other possible extension is that the clinic can assign the patients to the other days than their preferred appointment day in case each patient has a single preference. Patients either reject this assignment or accept it. The probability of accepting the appointment day offered by the clinic is higher for closer days to the patient's preferred day. We can define such a probability function to reflect this behavior of the patient. Secondly, it is assumed that patients always show up and they do not cancel their appointment. However, one can incorporate the situation that patients may not show up or cancel their appointment. Patients assigned to a closer day to her preferred appointment day have lower probability of no-show and cancellation. The clinic can resort to overbooking to compensate the no-shows. Moreover, cancellations and overtime work comes into the stage. Thirdly, it is possible to introduce a random service time instead of assuming that each appointment takes one appointment slot with the same length. Finally, we can have a separate constraint for each patient type, which gives a multiple-constrained MDP. All of these extensions can be considered in future research.

## Conflict of interest

The authors declare no conflict of interest.

## References

Ahmadi-Javid, A., Jalali, Z., & Klassen, K. J. (2017). Outpatient appointment systems in healthcare: A review of optimization studies. *European Journal of Operational Research*, 258(1), 3–34. Doi : https://dx.doi.org/10.1016/j.ejor.2016.06.064

Bowser, D. M., Utz, S., Glick, D., & Harmon, R. (2010). A systematic review of the relationship of diabetes mellitus, depression, and missed appointments in a low-income uninsured population. *Archives of psychiatric nursing*, 24(5), 317–329. Doi : https://doi.org/10.1016/j.apnu.2009.12.004

Cayirli, T., & Veral, E. (2003). Outpatient scheduling in health care: A review of literature. *Production and operations management*, 12(4), 519–549. Doi : https://doi.org/10.1111/j.1937-5956.2003.tb00218.x

Feldman, J., Liu, N., Topaloglu, H., & Ziya, S. (2014). Appointment scheduling under patient preference and no-show behavior. *Operations Research*, 62(4), 794 – 811. Doi : https://doi.org/10.1287/opre.2014.1286

Gocgun, Y., & 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. Doi : https://doi.org/10.1007/s10729-013-9253-z

Gupta, D., & Denton, B. (2008). Appointment scheduling in health care: Challenges and opportunities. *IIE transactions*, 40(9), 800–819. Doi : https://doi.org/10.1080/07408170802165880

Gupta, D., & Wang, L. (2008). Revenue management for a primary-care clinic in the presence of patient choice. *Operations Research*, 56(3), 576–592. Doi : https://doi.org/10.1287/opre.1080.0542

Magerlein, J. M., & Martin, J. B. (1978). Surgical demand scheduling: a review. *Health services research*, 13(4), 418. Retrieved from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1072083/pdf/hsresearch00545-0074.pdf

Parizi, M. S., & Ghate, A. (2016). Multi-class, multi-resource advance scheduling with no-shows, cancellations and overbooking. *Computers and Operations Research*, 67, 90–101. Doi : https://doi.org/10.1016/j.cor.2015.09.004

Patrick, J., Puterman, M. L., & Queyranne, M. (2008). Dynamic multipriority patient scheduling for a diagnostic resource. *Operations research*, 56(6), 1507– 1525. Doi : https://doi.org/10.1287/opre.1080.0590

Powell, W. B. (2011). *Approximate dynamic programming: Solving the curses of dimensionality*. New Jersey, USA: John Wiley and Sons.

Puterman, M. (1994). *Markov decision processes*. New Jersey, USA: John Wiley and Sons.

Saure, A., Patrick, J., Tyldesley, S., & Puterman, M. L. (2012). Dynamic multi-appointment patient scheduling for radiation therapy. *European Journal of Operational Research*, 223(2), 573–584. Doi : https://doi.org/10.1016/j.ejor.2012.06.046

Sennott, L. I. (1991). Constrained discounted markov decision chains. *Probability in the Engineering and Informational Sciences*, 5(4), 463–475. Doi : https://doi.org/10.1017/S0269964800002230

Sugiyama, M. (2015). *Statistical reinforcement learning: modern machine learning approaches*. Florida, USA: CRC Press.

Sutton, R. S., & Barto, A. G. (2011). *Reinforcement learning: An introduction*. Cambridge, USA: MIT Press

Truong, V.-A. (2015). Optimal advance scheduling. *Management Science*, 61(7), 1584–1597. Doi : https://doi.org/10.1287/mnsc.2014.2067

Wang, J., & Fung, R. Y. (2015). Dynamic appointment scheduling with patient preferences and choices. *Industrial Management and Data Systems*, 115(4), 700– 717. Doi : https://doi.org/10.1108/IMDS-12-2014-0372

Wang, W.Y., & Gupta, D. (2011). Adaptive appointment systems with patient preferences. *Manufacturing and Service Operations Management*, 13(3), 373–389. Doi : https://doi.org/10.1287/msom.1110.0332

## Appendix

*Satisfaction of Assumption 1:* Let $\psi$ be an arbitrary policy. We know that $H(\vec{s}_t, \vec{a}_t) \le \sum_{i=1}^{I} r_i C$ since total number of patients examined on day 0 cannot exceed $C$. Due to this fact and Equation (8), $K_\psi(\vec{s}) \le \frac{\sum_{i=1}^{I} r_i C}{1-\beta} < \infty$. This shows that $K(\vec{s})$ is finite for all policies. Similarly, by Equation (9) and the definition of $D(\vec{s}_t, \vec{a}_t)$, $C_\psi(\vec{s}) \le \frac{\sum_{i=1}^{I} \pi_i Y_i}{1-\beta} < \infty$, where $Y_i$ is the maximum number of type-$i$ patients that can arrive on a given day according to our state space definition. Therefore, our model satisfies Assumption 1.

*Proof of Proposition 1:* We need to show that the following inequality holds for every $(\vec{u}, \vec{x}) \in S$ and every combination of $i$ and $n$:

$$V_t^\theta(\vec{u}, \vec{x} + 2e_{in}) - 2V_t^\theta(\vec{u}, \vec{x} + e_{in}) + V_t^\theta(\vec{u}, \vec{x}) \le 0 \qquad \text{(A.1)}$$

We will use induction to show that (A.1) is satisfied for every $(\vec{u}, \vec{x}) \in S$ and every $(i, n)$. Let us call the system in state $\vec{s}_1 = (\vec{u}, \vec{x} + 2e_{in})$ as system A, the systems in state $\vec{s}_2 = \vec{s}_3 = (\vec{u}, \vec{x} + e_{in})$ as system B and C and the system in state $\vec{s}_4 = (\vec{u}, \vec{x})$ as system D.

Base case: We will prove that

$$V_0^\theta(\vec{u}, \vec{x} + 2e_{in}) - 2V_0^\theta(\vec{u}, \vec{x} + e_{in}) + V_0^\theta(\vec{u}, \vec{x}) \le 0 \qquad \text{(A.2)}$$

It can be easily seen by inspection that the optimal action in a given state $(\vec{u}, \vec{x})$ at $t = 0$ is

$a_{1n}^*(\vec{u}, \vec{x}) = \min\{u_{1n}, C - \sum_{l=1}^{I} x_{ln}\}$

$a_{1n}^*(\vec{u}, \vec{x}) = min\{u_{in}, C - \sum_{l=1}^{I} x_{ln} - \sum_{l=1}^{i-1} a_{ln}^*(\vec{u}, \vec{x})\} \;\; for\ i > 1\ and\ \forall n \in \mathcal{N}$

Under this action, it is obvious that either

$a_{in}^{A*}(\vec{s}_1) = a_{in}^{B*}(\vec{s}_2) = a_{in}^{C*}(\vec{s}_3) = a_{in}^{D*}(\vec{s}_4)$ or

$a_{in}^{A*}(\vec{s}_1) + 2 = a_{in}^{B*}(\vec{s}_2) + 1 = a_{in}^{C*}(\vec{s}_3) + 1 = a_{in}^{D*}(\vec{s}_4)$ hold.

The second case hold if there is not enough capacity on day $n$ of system A. In each case, we obtain the following equation:

$$\begin{aligned}
&V_0^\theta(\vec{u}, \vec{x} + 2e_{in}) - 2V_0^\theta(\vec{u}, \vec{x} + e_{in}) + V_0^\theta(\vec{u}, \vec{x}) \\
&= V_0^{\theta,A}(\vec{s}_1) - V_0^{\theta,B}(\vec{s}_2) - V_0^{\theta,C}(\vec{s}_3) + V_0^{\theta,D}(\vec{s}_4) \\
&= \left(\sum_{i=1}^{I} r_i(x_{i0} + a_{i0}^{A*}(\vec{s}_1)) - \sum_{i,n} \pi_i \left(u_{in} - a_{in}^{A*}(\vec{s}_1)\right)\right) - \left(\sum_{i=1}^{I} r_i\left(x_{i0} + a_{i0}^{B*}(\vec{s}_2)\right) - \sum_{i,n} \pi_i\left(u_{in} - a_{in}^{B*}(\vec{s}_2)\right)\right) \\
&\qquad - \left(\sum_{i=1}^{I} r_i \left(x_{i0} + a_{i0}^{C*}(\vec{s}_3)\right) - \sum_{i,n} \pi_i \left(u_{in} - a_{in}^{C*}(\vec{s}_3)\right)\right) \\
&\qquad + \left(\sum_{i=1}^{I} r_i(x_{i0} + a_{i0}^{D*}(\vec{s}_4)) - \sum_{i,n} \pi_i\left(u_{in} - a_{in}^{D*}(\vec{s}_4)\right)\right) \\
&= 0
\end{aligned}$$

Induction Hypothesis: Below inequality is satisfied for $t = 1, 2, \dots, t'$.

$$V_t^\theta(\vec{u}, \vec{x} + 2e_{i'n'}) - 2V_t^\theta(\vec{u}, \vec{x} + e_{i'n'}) + V_t^\theta(\vec{u}, \vec{x}) \le 0 \quad \forall i' \in \mathcal{I} \qquad \text{(A.3)}$$

We need to show that Inequality (A.3) is satisfied for $t = t' + 1$. Assume that the definitions for states $\vec{s}_1, \vec{s}_2, \vec{s}_3, \vec{s}_4$ and systems A, B, C and D are same as the base case. We let systems A and D follow the optimal policy. Without loss of generality, we may assume that there exists $m_{in} \in \{\dots, -1, 0, 1, \dots\}$ for every $i \neq i'$, $n \in \mathcal{N}$ and $m_{i'n'} \in$

$\{0, 1, \dots\}$ such that the relationship between the actions of system A in state $\vec{s}_1$ and system D in state $\vec{s}_4$ on day $n$ can be expressed as

$$a_{in}^{A*}(\vec{s}_1) + m_{in} = a_{in}^{D*}(\vec{s}_4)\ \forall i \in \mathcal{I}\ and\ \forall n \in \mathcal{N} \tag{A.4}$$

$m_{i'n'} \geq 0$ follows from the induction hypothesis. Let systems B and C in states $\vec{s}_2$ and $\vec{s}_3$ implement the following actions:

$$a_{in}^{B}(\vec{s}_2) = \begin{cases} a_{in}^{A*}(\vec{s}_1) + 1 & if\ i = i', n = n'\ and\ a_{in}^{A*}(\vec{s}_1) + m_{in} > 0 \\ a_{in}^{A*}(\vec{s}_1) & otherwise \end{cases} \tag{A.5}$$

where $x_{in}^{B}$ is the $x_{in}$ value of the state for system B.

$$a_{in}^{C}(\vec{s}_3) = \begin{cases} a_{in}^{A*}(\vec{s}_1) + m_{in} - 1 & if\ i = i', n = n'\ and\ a_{in}^{A*}(\vec{s}_1) + m_{in} > 0 \\ a_{in}^{A*}(\vec{s}_1) + m_{in} & otherwise \end{cases} \tag{A.6}$$

It is important to note that since systems A and B follow the optimal policy, their actions $a_{in}^{A*}(\vec{s}_1)$ and $a_{in}^{D*}(\vec{s}_4)$ are feasible. If they are feasible, then $a_{in}^{B}(\vec{s}_2)$ and $a_{in}^{C}(\vec{s}_3)$ given in Equation (A.5) and Equation (A.6) are also feasible. We inspect the following cases:

Case 1: If $n' = 0$ and $a_{i'n'}^{A*}(\vec{s}_1) + m_{i'n'} = 0$, none of the systems accept any patients. However, we will keep $a_{i'n'}^{A*}(\vec{s}_1)$ and $m_{i'n'}$ in our notation although both of them are zero. All systems reach the same state at $t = t' + 1$. (A.3) is satisfied as below:

$$V_t^{\theta}(\vec{u}, \vec{x} + 2e_{i'0}) - 2V_t^{\theta}(\vec{u}, \vec{x} + e_{i'0}) + V_t^{\theta}(\vec{u}, \vec{x})$$

$$\leq V_{t,A}^{\theta}(\vec{u}, \vec{x} + 2e_{i'0}) - V_{t,B}^{\theta}(\vec{u}, \vec{x} + e_{i'0}) - V_{t,C}^{\theta}(\vec{u}, \vec{x} + e_{i'0}) + V_{t,D}^{\theta}(\vec{u}, \vec{x})$$

$$= \Big[\textstyle\sum_{i \neq i'} r_i \left(x_{i0} + a_{i0}^{A*}(\vec{s}_1)\right) + r_{i'} \left(x_{i'0} + 2 + a_{i'0}^{A*}(\vec{s}_1)\right) - \theta \sum_{i \neq i'} \pi_i (u_{i0} - a_{i0}^{A*}(\vec{s}_1))$$

$$-\theta \pi_{i'} \left(u_{i'0} - a_{i'0}^{A*}(\vec{s}_1)\right) - \theta \textstyle\sum_{i=1}^{I} \sum_{n=1}^{N} \pi_i \left(u_{in} - a_{in}^{A*}(\vec{s}_1)\right) + \beta \sum_{\vec{y} \in \mathbb{Y}} P(\vec{y}) V(\vec{y}, \vec{x}')\Big]$$

$$-\Big[\textstyle\sum_{i \neq i'} r_i \left(x_{i0} + a_{i0}^{A*}(\vec{s}_1)\right) + r_{i'} \left(x_{i'0} + 1 + a_{i'0}^{A*}(\vec{s}_1)\right) - \theta \sum_{i \neq i'} \pi_i (u_{i0} - a_{i0}^{A*}(\vec{s}_1))$$

$$-\theta \pi_{i'} \left(u_{i'0} - a_{i'0}^{A*}(\vec{s}_1)\right) - \theta \textstyle\sum_{i=1}^{I} \sum_{n=1}^{N} \pi_i \left(u_{in} - a_{in}^{A*}(\vec{s}_1)\right) + \beta \sum_{\vec{y} \in \mathbb{Y}} P(\vec{y}) V(\vec{y}, \vec{x}')\Big]$$

$$-\big[\textstyle\sum_{i \neq i'} r_i (x_{i0} + a_{i0}^{A*}(\vec{s}_1) + m_{i0}) + r_{i'}\big(x_{i'0} + 1 + a_{i'0}^{A*}(\vec{s}_1) + m_{i'0}\big) - \theta \sum_{i \neq i'} \pi_i (u_{i0} - a_{i0}^{A*}(\vec{s}_1) -$$

$$m_{i0}) - \theta \pi_{i'}\big(u_{i'0} - a_{i'0}^{A*}(\vec{s}_1) - m_{i'0}\big) - \theta \textstyle\sum_{i=1}^{I} \sum_{n=1}^{N} \pi_i (u_{in} - a_{in}^{A*}(\vec{s}_1) - m_{in}) + \quad \beta \sum_{\vec{y} \in \mathbb{Y}} P(\vec{y}) V(\vec{y}, \vec{x}')\big]$$

$$+\big[\textstyle\sum_{i \neq i'} r_i (x_{i0} + a_{i0}^{A*}(\vec{s}_1) + m_{i0}) + r_{i'}\big(x_{i'0} + a_{i'0}^{A*}(\vec{s}_1) + m_{i'0}\big) - \theta \sum_{i \neq i'} \pi_i (u_{i0} - a_{i0}^{A*}(\vec{s}_1) - m_{i0})$$

$$-\theta \pi_{i'}\big(u_{i'0} - a_{i'0}^{A*}(\vec{s}_1) - m_{i'0}\big) - \theta \textstyle\sum_{i=1}^{I} \sum_{n=1}^{N} \pi_i (u_{in} - a_{in}^{A*}(\vec{s}_1) - m_{in}) + \beta \sum_{\vec{y} \in \mathbb{Y}} P(\vec{y}) V(\vec{y}, \vec{x}')\big]$$

$$= 0$$

Case 2: If $n' = 0$ and $a_{i'n'}^{A*}(\vec{s}_1) + m_{i'n'} > 0$, all systems reach the same state at $t = t' + 1$ by the transition in Equation (6). (A.3) is satisfied as below:

$$V_t^{\theta}(\vec{u}, \vec{x} + 2e_{i'0}) - 2V_t^{\theta}(\vec{u}, \vec{x} + e_{i'0}) + V_t^{\theta}(\vec{u}, \vec{x})$$

$$\leq V_{t,A}^{\theta}(\vec{u}, \vec{x} + 2e_{i'0}) - V_{t,B}^{\theta}(\vec{u}, \vec{x} + e_{i'0}) - V_{t,C}^{\theta}(\vec{u}, \vec{x} + e_{i'0}) + V_{t,D}^{\theta}(\vec{u}, \vec{x})$$

$$= \Big[\sum_{i\neq i'} r_i \left(x_{i0} + a_{i0}^{A*}(\vec{s}_1)\right) + r_{i'} \left(x_{i'0} + 2 + a_{i'0}^{A*}(\vec{s}_1)\right) - \theta \sum_{i\neq i'} \pi_i (u_{i0} - a_{i0}^{A*}(\vec{s}_1))$$

$$-\theta\pi_{i'} \left(u_{i'0} - a_{i'0}^{A*}(\vec{s}_1)\right) - \theta \sum_{i=1}^{I} \sum_{n=1}^{N} \pi_i \left(u_{in} - a_{in}^{A*}(\vec{s}_1)\right) + \beta \sum_{\vec{y}\in\mathbb{Y}} P(\vec{y}) V(\vec{y},\vec{x}')\Big]$$

$$- \Big[\sum_{i\neq i'} r_i \left(x_{i0} + a_{i0}^{A*}(\vec{s}_1)\right) + r_{i'}\left(x_{i'0} + 1 + a_{i'0}^{A*}(\vec{s}_1) + 1\right) - \theta \sum_{i\neq i'} \pi_i (u_{i0} - a_{i0}^{A*}(\vec{s}_1))$$

$$-\theta\pi_{i'}\left(u_{i'0} - a_{i'0}^{A*}(\vec{s}_1) - 1\right) - \theta \sum_{i=1}^{I} \sum_{n=1}^{N} \pi_i \left(u_{in} - a_{in}^{A*}(\vec{s}_1)\right) + \beta \sum_{\vec{y}\in\mathbb{Y}} P(\vec{y}) V(\vec{y},\vec{x}')\Big]$$

$$-[\sum_{i\neq i'} r_i (x_{i0} + a_{i0}^{A*}(\vec{s}_1) + m_{i0}) + r_{i'}\left(x_{i'0} + 1 + a_{i'0}^{A*}(\vec{s}_1) + m_{i'0} - 1\right) - \theta \sum_{i\neq i'} \pi_i (u_{i0} - a_{i0}^{A*}(\vec{s}_1) - m_{i0}) - \theta\pi_{i'}\left(u_{i'0} - a_{i'0}^{A*}(\vec{s}_1) - m_{i'0} + 1\right) - \theta \sum_{i=1}^{I} \sum_{n=1}^{N} \pi_i (u_{in} - a_{in}^{A*}(\vec{s}_1) - m_{in}) + \beta \sum_{\vec{y}\in\mathbb{Y}} P(\vec{y}) V(\vec{y},\vec{x}')]$$

$$+[\sum_{i\neq i'} r_i (x_{i0} + a_{i0}^{A*}(\vec{s}_1) + m_{i0}) + r_{i'}\left(x_{i'0} + a_{i'0}^{A*}(\vec{s}_1) + m_{i'0}\right) - \theta \sum_{i\neq i'} \pi_i (u_{i0} - a_{i0}^{A*}(\vec{s}_1) - m_{i0})$$

$$-\theta\pi_{i'}\left(u_{i'0} - a_{i'0}^{A*}(\vec{s}_1) - m_{i'0}\right) - \theta \sum_{i=1}^{I} \sum_{n=1}^{N} \pi_i (u_{in} - a_{in}^{A*}(\vec{s}_1) - m_{in}) + \beta \sum_{\vec{y}\in\mathbb{Y}} P(\vec{y}) V(\vec{y},\vec{x}')]$$

$$= 0$$

Case 3: If $n' > 0$ and $a_{i'n'}^{A*}(\vec{s}_1) + m_{i'n'} = 0$, none of the systems accept any patients. (A.3) is satisfied as below:

$$V_t^\theta(\vec{u},\vec{x} + 2e_{i'0}) - 2V_t^\theta(\vec{u},\vec{x} + e_{i'0}) + V_t^\theta(\vec{u},\vec{x})$$

$$\leq V_{t,A}^\theta(\vec{u},\vec{x} + 2e_{i'0}) - V_{t,B}^\theta(\vec{u},\vec{x} + e_{i'0}) - V_{t,C}^\theta(\vec{u},\vec{x} + e_{i'0}) + V_{t,D}^\theta(\vec{u},\vec{x})$$

$$= [\sum_{i\neq i'} r_i x_{i0} + r_{i'}(x_{i'0}) - \theta \sum_{i\neq i'} \pi_i u_{i0} - \theta\pi_{i'} u_{i'0} - \theta \sum_{i=1}^{I} \sum_{n=1}^{N} \pi_i u_{in}$$

$$+\beta \sum_{\vec{y}\in\mathbb{Y}} P(\vec{y}) V(\vec{y},\vec{x}'' + 2e_{i'(n'-1)})]$$

$$-[\sum_{i\neq i'} r_i x_{i0} + r_{i'}(x_{i'0}) - \theta \sum_{i\neq i'} \pi_i u_{i0} - \theta\pi_{i'} u_{i'0} - \theta \sum_{i=1}^{I} \sum_{n=1}^{N} \pi_i u_{in}$$

$$+\beta \sum_{\vec{y}\in\mathbb{Y}} P(\vec{y}) V(\vec{y},\vec{x}'' + e_{i'(n'-1)})]$$

$$-[\sum_{i\neq i'} r_i x_{i0} + r_{i'}(x_{i'0}) - \theta \sum_{i\neq i'} \pi_i u_{i0} - \theta\pi_{i'} u_{i'0} - \theta \sum_{i=1}^{I} \sum_{n=1}^{N} \pi_i u_{in}$$

$$+\beta \sum_{\vec{y}\in\mathbb{Y}} P(\vec{y}) V(\vec{y},\vec{x}'' + e_{i'(n'-1)})]$$

$$+\Big[\sum_{i\neq i'} r_i x_{i0} + r_{i'}(x_{i'0}) - \theta \sum_{i\neq i'} \pi_i u_{i0} - \theta\pi_{i'} u_{i'0} - \theta \sum_{i=1}^{I} \sum_{n=1}^{N} \pi_i u_{in} + \beta \sum_{\vec{y}\in\mathbb{Y}} P(\vec{y}) V(\vec{y},\vec{x}'')\Big]$$

$$\leq 0$$

where the last inequality follows from the induction hypothesis.

Case 4: If $n' > 0$ and $a_{i'n'}^{A*}(\vec{s}_1) + m_{i'n'} > 0$, system B reaches the same state as system A at $t = t' + 1$. Similarly, system C reaches the same state as system D at $t = t' + 1$ and (A.3) is satisfied as below:

$$V_t^\theta(\vec{u},\vec{x} + 2e_{i'0}) - 2V_t^\theta(\vec{u},\vec{x} + e_{i'0}) + V_t^\theta(\vec{u},\vec{x})$$

$$\leq V_{t,A}^\theta(\vec{u},\vec{x} + 2e_{i'0}) - V_{t,B}^\theta(\vec{u},\vec{x} + e_{i'0}) - V_{t,C}^\theta(\vec{u},\vec{x} + e_{i'0}) + V_{t,D}^\theta(\vec{u},\vec{x})$$

$$= \Big[\sum_{i\neq i'} r_i \left(x_{i0} + a_{i0}^{A*}(\vec{s}_1)\right) + r_{i'} \left(x_{i'0} + a_{i'0}^{A*}(\vec{s}_1)\right) - \theta \sum_{i\neq i'} \pi_i (u_{i0} - a_{i0}^{A*}(\vec{s}_1))$$

$$-\theta\pi_{i'} \left(u_{i'0} - a_{i'0}^{A*}(\vec{s}_1)\right) - \theta \sum_{i=1}^{I} \sum_{n=1}^{N} \pi_i \left(u_{in} - a_{in}^{A*}(\vec{s}_1)\right) + \beta \sum_{\vec{y}\in\mathbb{Y}} P(\vec{y}) V(\vec{y},\vec{x}')\Big]$$

$$- \Big[\sum_{i\neq i'} r_i \left(x_{i0} + a_{i0}^{A*}(\vec{s}_1)\right) + r_{i'} \left(x_{i'0} + a_{i'0}^{A*}(\vec{s}_1)\right) - \theta \sum_{i\neq i'} \pi_i (u_{i0} - a_{i0}^{A*}(\vec{s}_1))$$

$$-\theta\pi_{i'} \left(u_{i'0} - a_{i'0}^{A*}(\vec{s}_1)\right) - \theta \sum_{i=1}^{I} \sum_{n=1}^{N} \pi_i \left(u_{in} - a_{in}^{A*}(\vec{s}_1)\right) + \beta \sum_{\vec{y}\in\mathbb{Y}} P(\vec{y}) V(\vec{y},\vec{x}')\Big]$$

$$-[\sum_{i\neq i'} r_i (x_{i0} + a_{i0}^{A*}(\vec{s}_1) + m_{i0}) + r_{i'}\left(x_{i'0} + a_{i'0}^{A*}(\vec{s}_1) + m_{i'0}\right) - \theta \sum_{i\neq i'} \pi_i (u_{i0} - a_{i0}^{A*}(\vec{s}_1) - m_{i0}) - \theta\pi_{i'}\left(u_{i'0} - a_{i'0}^{A*}(\vec{s}_1) - m_{i'0}\right) - \theta \sum_{i=1}^{I} \sum_{n=1}^{N} \pi_i (u_{in} - a_{in}^{A*}(\vec{s}_1) - m_{in}) + \beta \sum_{\vec{y}\in\mathbb{Y}} P(\vec{y}) V(\vec{y},\vec{x}'')]$$

$$+\left[\sum_{i\neq i'} r_i(x_{i0}+a_{i0}^{A*}(\vec{s}_1)+m_{i0})+r_{i'}\left(x_{i'0}+a_{i'0}^{A*}(\vec{s}_1)+m_{i'0}\right)-\theta\sum_{i\neq i'}\pi_i(u_{i0}-a_{i0}^{A*}(\vec{s}_1)-m_{i0})\right.$$
$$\left.-\theta\pi_{i'}\left(u_{i'0}-a_{i'0}^{A*}(\vec{s}_1)-m_{i'0}\right)-\theta\sum_{i=1}^{I}\sum_{n=1}^{N}\pi_i(u_{in}-a_{in}^{A*}(\vec{s}_1)-m_{in})+\beta\sum_{\vec{y}\in\mathbb{Y}}P(\vec{y})V(\vec{y},\vec{x}'')\right]$$
$$=0$$

Furthermore, the following two equations also hold for our model:

$$V^{\theta}(\vec{u},\vec{x}+e_{i'n'}+e_{i''n''})-V^{\theta}(\vec{u},\vec{x}+e_{i'n'})-V^{\theta}(\vec{u},\vec{x}+e_{i'''n'''}+e_{i''n''})+V^{\theta}(\vec{u},\vec{x}+e_{i'n'})=0 \tag{A.7}$$

$$V^{\theta}(\vec{u},\vec{x}+e_{i'n'}+e_{i'n''})-V^{\theta}(\vec{u},\vec{x}+e_{i'n'})-V^{\theta}(\vec{u},\vec{x}+e_{i'n''})+V^{\theta}(\vec{u},\vec{x})=0 \tag{A.8}$$

The proofs of the Equations (A.7) and (A.8) is very similar to the concavity proof. Therefore, we skip these proofs. The optimality of the "strict booking limit policy" results from the concavity of the value function, Equation (A.7) and (A.8).

*Satisfaction of Assumption 2:* Let $\Phi$ be a policy that rejects all customers. Since Assumption 1 is satisfied, $K(\vec{s})<\infty$ holds under each policy. From the proof of Proposition 1, it can be concluded that the booking limit policy minimizes the expected discounted rejection cost. Let $\Phi$ the best booking limit policy for the expected discounted rejection cost. If $c$ is set by the user such that $C_{\Phi}(\vec{s})<c$, then Assumption 2 is satisfied.

*Proof of Theorem 1:* The proof is the direct result of Sennott (1991). Our model always satisfies Assumption 1 and it also satisfies Assumption 2 given that $c$ is set conveniently. Our action space, $\mathbb{A}=\prod_{\vec{s}\in S}A_{\vec{s}}$, is compact since it is finite. Let $\gamma=\inf\{\theta>0|C^{\theta}(\vec{s})\leq c\}$. By Lemma 3.8 of Sennott (1991), $\gamma<\infty$.

**Case 1:** $\gamma>0$:

Case 1a: If there exists a $\gamma$-optimal stationary policy $f$ such that $C_f^{\theta}(\vec{s})\leq c$, by Lemma 3.7 of Sennott (1991), $f$ optimally solves the constrained problem. Since $f$ is a $\gamma$-optimal policy, it is a booking limit policy by Proposition 1.

Case 1b: Since $\mathbb{A}$ is compact, we may find a sequence $\theta_n\downarrow\gamma$ and a stationary policy $f$ such that $f(\theta_n)\rightarrow f$. There is also a sequence $\delta_n\uparrow\gamma$ and a stationary policy e such that $f(\delta_n)\rightarrow e$. Since $f(\theta_n)$ is $\theta_n$-optimal policy (similarly $f(\delta_n)$ is $\delta_n$-optimal policy), $f(\theta_n)$ and $f(\delta_n)$ are sequences of booking limit policies by Proposition 1. By the definition of convergence, they converge to a booking limit policy. Therefore, $f$ and $e$ are booking limit policies. By Lemma 3.6 of Sennott (1991), $f$ and $e$ are $\gamma$-optimal. Furthermore, by Lemma 3.6 of Sennott (1991), $C_{f(\theta_n)}(\vec{s})\rightarrow C_f(\vec{s})$ and $C_{f(\delta_n)}(\vec{s})\rightarrow C_e(\vec{s})$. By the definition of $\gamma$, $C_f(\vec{s})<c$ and $C_e(\vec{s})>c$. By Lemma 3.9 of Sennott (1991), the randomized policy $(p,f,e)$ is $\gamma$-optimal since both $f$ and $e$ are $\gamma$-optimal. Because by Lemma 3.9 of Sennott (1991), $C_{(p,f,e)}(\vec{s})$ is a continuous function of $p$ for all $\vec{s}\in S$, we may choose $p$ such that $C_{(p,f,e)}(\vec{s})=c$ since $C_f(\vec{s})<c$ and $C_e(\vec{s})>c$.

Then the optimal policy is the mixture of two booking limit policies. Our claim that there exists at most one state for which those two stationary policies differ directly follows from the proof of Theorem 2.1 in Sennott (1991).

**Case 2:** $\gamma=0$:

There exists a sequence $\theta_n\downarrow\gamma$ and a stationary policy $f$ such that $f(\theta_n)\rightarrow f$. Since $f(\theta_n)$ is $\theta_n$-optimal policy, $f(\theta_n)$is a sequence of booking limit policies by Proposition 1. By the definition of convergence, $f(\theta_n)$ converges to a booking limit policy. Therefore, $f$ is also a booking limit policy. It directly follows from the proof of Theorem 2.1 in Sennott (1991) that $f$ optimally solves the constrained MDP.