J Comb Optim (2019) 37:123–149
https://doi.org/10.1007/s10878-017-0216-z

Online scheduling for outpatient services with heterogeneous patients and physicians

Huiqiao Su[1] · **Guohua Wan**[1] · **Shan Wang**[1]

Published online: 28 November 2017

Abstract In outpatient services, it is critical to schedule patients for physicians to reduce both patients waiting and physicians overtime working. In this paper, we regard the problem as an online scheduling problem and based on analysis of a real data set from a big hospital in China, we develop a dynamic programming model to solve the problem. We propose a Policy Iteration Algorithm to find the optimal solution in the steady state, and obtain the structural properties of the policy. We conduct numerical experiments to compare the performance of the policy with that of the two policies used in practice by simulating various scenarios. The numerical results show that the policy has the best performance across all scenarios, especially when the system is heavily loaded. We also discuss the managerial implications of the study for practitioners. The model and solution method can be easily extended to multi-server case and can be applied to the general service scheduling problems with heterogeneous customers and service providers.

Keywords Health care operations · Online scheduling · Scheduling policy · Markov decision process

This research work is supported in part by NSF of China (Grant Nos. 71520107003 and 71421002).

✉ Guohua Wan
ghwan@sjtu.edu.cn

[1] Antai College of Economics and Management, Shanghai Jiao Tong University, Shanghai 200030, China

1 Introduction

In hospitals, the physicians can generally be classified into two categories: junior ones and senior ones. A senior physician is able to handle both the normal and the complicated diseases, while a junior physician generally handles the normal ones. However, it is often the case that the patients with ailments would like to wait for the senior physicians. This causes long waiting time for the patients resulting in unsatisfaction, as well as the high workload for the senior physicians which may affect the service quality (due to pressure from the patients, which leads to physician's control of the consulting times).

As the largest developing country in the world, China is facing the most critical challenges in health care. In particular, the outpatients scheduling problem is more critical in China than in many countries. One reason is that in China, people normally do not make appointment for the services, and the patients are free to choose the hospitals as well as the physicians. Another reason is the mismatch and serious waste of health care resources. Even with very general diseases, the patients still prefer waiting to see the senior physicians. Hence we can see that many patients are waiting anxiously for the senior physicians who probably have to work overtime but the junior physicians may be idle. This causes the patients complaint of long waiting time but the patients with serious diseases cannot be serviced timely. Professor Burns from Wharton shares his opinions in Knowledge@Wharton (2013) towards the issues of China's health care system: "everyone … goes to … major academic health centers … enormous lines … to see a specialist", and "people expect good care … when they don not receive it, they blame the physicians".

To improve such a situation, some efficient systems are developed to guide the patients and make the use of the health care resource efficiently. For example, the "Gatekeeper" system in UK requires the patients to have medical examinations in the community health care centers before they are transfered to high level hospitals. Inspired by the "Gatekeeper" system, we study the policy design for the hospital outpatient services to assign the patients to the physicians, taking into consideration the status of the patients and the whole service system.

According to the interviews with Doctor Ge (2016), the director of outpatient services in Shanghai General Hospital, one of the largest hospitals in China, the scheduling process in Chinese hospitals can be depicted as in Fig. 1.

As can be seen in Fig. 1, when the patients arrive, they register with the registration office first. Then they go to the scheduling center of the relevant department, where the patients obtain the information of the physicians on duty. After that, the

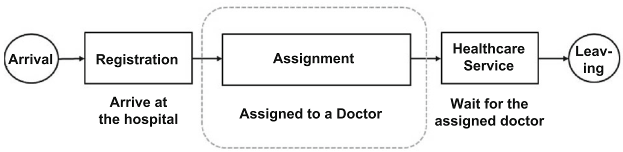

Fig. 1 Flow chart of the outpatient services

patients have two options: choosing the physician they prefer ("Free to Choose Policy", and it is highly likely that they choose the senior physicians), or letting the system to assign them to any physicians, where the patients are assigned to the physician with the shortest waiting time ("Myopic Policy"). Once scheduled, the patients wait for the assigned physician and normally they cannot be reassigned to other physicians.

The current process has quite a few drawbacks, according to Doctor Ge (2016). The major problem is the mismatch of resources: if the patients are free to choose, they choose the senior physicians in the most cases, even their diseases are normal, and then wait longer than if they choose junior physicians (without loss of quality of services). This results in that the complicated patients who really need the seniors cannot be serviced timely and that the service quality of the senior physicians may be sacrificed due to pressure of many waiting patients, and the seniors physicians may have to work overtime.

This phenomenon not only occurs in health care services, but also in many other service sectors. In a service system, there may be servers of different levels. Without loss of generality, we assume there are two levels: normal servers and advanced servers. The advanced servers are able to handle both the normal customers and demanding customers, while the normal servers can only handle the normal customers. Furthermore, assume that the cost for the same customers are the same no matter who services them. Often, customers like to wait for the advanced servers even with normal demand, resulting in the long waiting time for the customers, in particular for the high demanding customers who have to wait for the advanced servers. In this paper, we study how to design the scheduling policy for the services, taking into consideration the heterogeneity of both customers and servers. This problem rises in many applications such as repair services and banking services.

This study is motivated by the challenges faced by the outpatient service management where the heterogeneous patients come for healthcare services without appointment. The challenges become more evident in the analysis of a large data set obtained from Shanghai General Hospital. To deal with the issues, we focus on the scheduling policy design for online outpatient scheduling, assuming that the patients arrival is online and the processing time is stochastic. The objective is to find a scheduling policy for this online scheduling problem to minimize the waiting times of served patients and the number of deferred patients (in the form of penalty cost) when the service system is heavily loaded.

Specifically, we address the following questions:

1. For this online scheduling problem, are "Free to Choose Policy" or the currently used "Myopic Policy" good enough?
2. Can we get a better policy?

The goal of an effective scheduling tool is to match demands with capacity so that resources are fully utilized and the patients waiting times are minimized. To be practical, we consider two important characteristics in the scheduling problem: online arrivals of patients and stochastic service times, and the heterogeneity of physicians and patients.

In summary, the contributions of this study are three-folds:

1. We formally describe the whole process of outpatient services in China, and model it as an online scheduling problem. We show that a hospital does need a scheduling center to assign the patients to physicians for better performance.
2. We propose a new scheduling policy based on Markov decision process, and show that this policy outperforms both the "Myopic Policy" and the "Free to Choose Policy" by intensive numerical experiments over a fairly broad setting of parameters.
3. By developing an approach based Markov decision process, we can handle randomness in the outpatient scheduling problems, and it can be extended to other scheduling problems in different applications.

The remainder of this paper is organized as follows. In Sect. 2, we present a review of literature. In Sect. 3, we construct an online scheduling model for the problem and develop MDP-based policy based on empirical studies. The sensitivity analysis of parameters and performance evaluation of different policies are provided in Sect. 4. Finally, we summarize and discuss the extension in Sect. 5.

2 Literature review

The online outpatient scheduling problem is related to three streams of literature, namely, the outpatients scheduling with walk-ins, the online scheduling problems, and the applications of MDP in health care management.

2.1 Outpatients scheduling

The literature on outpatient scheduling is extensive and most of the them focuses on the appointment scheduling for outpatient services, without considering the walk-in patients. In this vein, there are two streams of literature. Gupta and Denton (2008), Hassin and Mendel (2008), Begen and Queyranne (2011) and Kong et al. (2013) study the problems with exact appointment time for each patient, while Kaandorp and Koole (2007), LaGanga and Lawrence (2012) and Zacharias and Pinedo (2014) study the problem for deciding the number of patients assigned to each slot. However, the presence of walk-in patients cannot be neglected since it makes big differences on the performance of schedules, see Cayirli and Veral (2003) for more discussions. When walk-in patients are considered, there is only a little literature, for example, Fetter and Thompson (1966), Rising et al. (1973), Cayirli et al. (2006), Cayirli et al. (2012) and Wang et al. (2016). All of these studies focus on the offline scheduling problems.

In addition, Qu et al. (2015) and Wiesche et al. (2016) study how to use the service capacity when walk-in patients accumulate. Green et al. (2006) study the problem of managing patient services in a diagnostic medical facility, where the patients include the outpatients, the inpatients and the emergency outpatients. They design the appointment schedule and find the dynamic priority rules for admitting outpatients into services.

2.2 Online scheduling

Online scheduling is a sub-area in scheduling theory. In online scheduling problems, customers (jobs) arrive over time, see Tan and Zhang (2013) for a comprehensive survey.

In case some parameters of an online scheduling problem are random, then the problem becomes an online stochastic scheduling problem. Chou et al. (2006) study a single machine online stochastic scheduling problem to minimize total weighted completion time with release time. They propose a rule called the "Weighted Shortest Processing Time among Available jobs (WSEPTA)" and perform an asymptotic analysis. Megow et al. (2006) combine the main features of online and stochastic scheduling on a single machine in a simple and natural way. They analyze a simple, combinatorial online scheduling policy and derive performance guarantee. For multiple machines, Chen and Shen (2007) analyze a stochastic online scheduling problem to minimize the total weighted completion times. They show that any non-idling policy is asymptotically optimal. Schulz (2008) presents a randomized online policy that achieves a bound of $2 + \delta$, and it is a fixzed scheduling policy.

Many studies consider online scheduling with preemption. For single machine, Sevcik (1974) introduces a priority policy that relies on an index computed for each job based on the properties of this job. Weiss (1995) formulates Sevcik's priority index based on Gittins index, called Gittins Index Priority Policy (GIPP), and provides a different proof for the optimality of the priority policy. Megow and Vredeveld (2006) formulate two variants of GIPP that work online. Assuming preemption is allowed, Megow and Vredeveld (2006) extend the F-GIPP to identical parallel machine case: at any time, process the m first jobs in the list of job pieces, or if there are less than m uncompleted jobs, process all uncompleted jobs. They show that it is a 2-approximation policy.

For applications of online scheduling in health care management, Berg and Denton (2017) study the health care delivery problem for dynamically allocating patients to procedure rooms in outpatient procedure centers. They model the problem as online stochastic extensible bin packing with the objective to minimize the total costs of opening procedure rooms and overtime working to complete the procedures in a day. They obtain both theoretical performance guarantees and average case performance and conduct numerical experiments.

2.3 Markov decision process in healthcare management

Markov decision process (MDP) method has wide applications in health care management. Schaefer et al. (2005) present an overview of MDP models and solution techniques in the context of medical decision-making. Hu et al. (1996), Ahn and Hornberger (1996) and Magni et al. (2000) are examples of successful applications of MDPs in medical decision-making. Sloan (2007) formulate an MDP model to study the tradeoffs involving in safety and cost in medical device reuse. Bennett and Hauser (2013) develop a general purpose (non-disease-specific) computational/artificial intelligence (AI) framework to make optimal service decisions. Combining MDP and dynamic decision networks for learning from clinical data, they develop complex plans via simulation of alternative sequential decision paths. They demonstrate that the proposed AI simulation framework can approximate optimal decisions even in complex and uncertain environments.

MDP also has applications in health care management. Kolesar (1970) propose an MDP model to deal with prescheduling of elective admissions. Patrick (2012) develop an MDP model to manage significant no-show rates and escalating appointment lead times in an outpatient clinic, and show that the MDP policy is as good as, or better than the open access ("do today's demand today") by simulation over a wide variety of potential scenarios and clinics, in terms of minimizing costs (or maximizing profits) as well as providing more consistent throughput. Li et al. (2015) formulate a Markov decision process model to optimize the colorectal cancer screening program, and design a screening schedule which balances the risk of colorectal cancer and the side-effects of colonoscopy, by taking the patients personal characteristics and compliances into account.

Unlike outpatient scheduling literature mentioned in Sect. 2.1, we are interested in dealing with a large-scale scheduling problem in current practices—the outpatient scheduling problem with heterogeneous patients as well as physicians. Due to the variety of outpatients' demands and different physician seniority, there may be significant waste of healthcare resources and long waiting time of outpatients which, of course, lower the satisfactory level of outpatients. However, it is difficult to obtain an optimal appointment schedule for large scale problems via optimization. Thus, a good scheduling policy which dynamically schedules the outpatients for physicians is necessary. Compared with the literature on online scheduling problems (see Sect. 2.2), the "jobs" and "machines" in our problem are heterogeneous, and the arrival process and service process are stochastic, hence, to obtain a proper policy for the online stochastic scheduling problem is quite challenging, and a new way is to study policy design for the online stochastic scheduling problem using MDP.

In summary, we can see that there is no prior modeling work on online stochastic scheduling that takes into consideration of policy design with heterogeneity of patients and physicians, where the solutions are obtained by using MDP method. This is exactly the problem and solution method presented in this paper.

3 Problem formulation

3.1 Problem description

The online stochastic outpatients scheduling problam can be described as follows. Without loss of generality, assuming that there are two categories of physicians: the junior physician, denoted by D_J, and the senior physician, denoted by D_S. Similarly, there are two types of patients, normal patients, P_G, and complicated patients, P_C. According to Fig. 1, once a patient registers, a physician is assigned to him or her. A normal patient P_G may be serviced by any physician, while a complicated patients must be be serviced by a senior physician. Because of the randomness of the patients' arrival process and service process, assigning many normal patients to D_S may cause long waiting for D_S, and vice versa. Besides the cost of patients waiting, there is another cost called "deferment cost". If the number of patients waiting for a physician exceeds the predetermined capacity, then the new arrivals have to be deferred to the next day, incurring a penalty cost. The objective is to minimize the total cost of waiting and deferment. Note that in this problem, the type of the next patient is unknown, and his/her arrival time and service time are random.

3.2 Empirical analysis of arrival and service processes

Before we elaborate on how to model and solve the problem, we exploit a real data set to figure out the arrival process of the patients and the service process of the physician. The data extracted from the EMR (Electronic Medical Records) system of Shanghai General Hospital records the patient visits in the department of gynecology, and spans for 1 year from October 2015 to September 2016. It contains the registration time and the service starting time for each visit during the time period, resulting in over 80k records in total. In the data set, the registration time can be used to represent the arrival time but the service time of each patient are not available directly from the data set. However, we can utilize the number of served patients during a certain time interval to analyze the service process. There are 23 physicians in the data set, with the average number of served patients by a physician per day is around 32. To eliminate outliers, we focus on the physicians who serve 10–50 patients per day, resulting in 16 physicians. We take the morning session for analysis since the physicians have lunch breaks at 11am. To study the service process, we use the data of which service starting time is among 8–10:30 am (the office hour starts at 8 am). For arrival process, we focus on the session from 7:30 to 10 am. This is because the data shows that the number of patients who register before the office hour is not negligible. More details of the data are presented in the following sections.

3.2.1 Arrival process

Most literature suggests that the patients arrival process follows a Poisson process (e.g., Swartzman 1970; Kim and Whitt 2014) and it is also observed in our data. We divide the whole two and half hour session into 30 slots equally, and count the registration number in each slot. Figure 2 shows the histograms of these counts. We can see that Poisson distribution is a good fit for the arrival process. Hence we propose the following hypothesis:

Null Hypothesis (H_0) The number of arrivals in each slot follows a Poisson distribution.
Aternative Hypothesis (H_1) The number of arrivals in each slot does not follow a Poisson distribution.
We test the above hypothesis for each physician to check if the arriving process is Poisson or not.

Small p-value in the statistical analysis rejects the Null Hypothesis. The p-values in Table 1 shows the goodness of fitting. Among the total 16 data subsets, the p-values

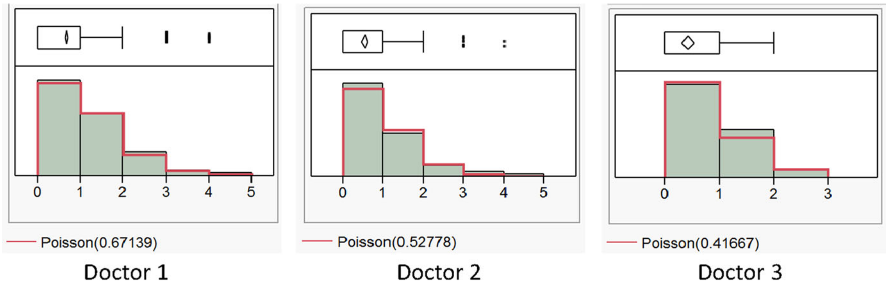

Fig. 2 Histograms of the registered patients in each time slot

Table 1 Statistical results for arrival process

Doctor	Mean	SD	SE	χ^2	Prob > χ^2
SL	0.41	0.66	0.01	7.24	0.20
ZS	0.48	0.72	0.02	8.93	0.11
QQ	1.16	1.12	0.03	9.26	0.23
GX	0.12	0.35	0.01	1.33	0.72
XW	0.60	0.80	0.03	7.11	0.21
YY	0.90	0.97	0.04	3.46	0.75
YZ	0.31	0.56	0.01	2.51	0.64
YC	0.36	0.62	0.01	11.45	0.08*
WX	0.48	0.71	0.01	8.63	0.13
LC	0.49	0.71	0.02	3.43	0.75
YM	0.90	1.06	0.02	76.22	0.00***
TS	0.34	0.59	0.02	2.78	0.73
YF	0.66	0.84	0.02	11.15	0.08*
MQ	0.23	0.49	0.01	10.33	0.07*
WH	0.72	0.85	0.02	3.92	0.69
JZ	0.54	0.77	0.01	12.25	0.06*

*$0.5 < p$-value ≤ 0.1;
***p-value < 0.01

of 11 subsets are larger than 0.1, and only the p-value of 1 subset is too small to reject the Null Hypothesis. These empirical results indicate that the counts of arrivals follow the Poisson distribution in most cases. Hence, it is reasonable to assume that the arrival process is a Poisson process.

3.2.2 Service process

As mentioned before, the service times are not available directly from the data set. However, we can count the number of patients who are served during a small interval (e.g., 5-min interval), then use these counts to analyze the service process. The histograms for these counts are shown in Fig. 3. It indicates that these counts follow Poisson distribution, meaning that the service time is exponentially distributed. There are also empirical evidences that clinic service times follow exponential distribution

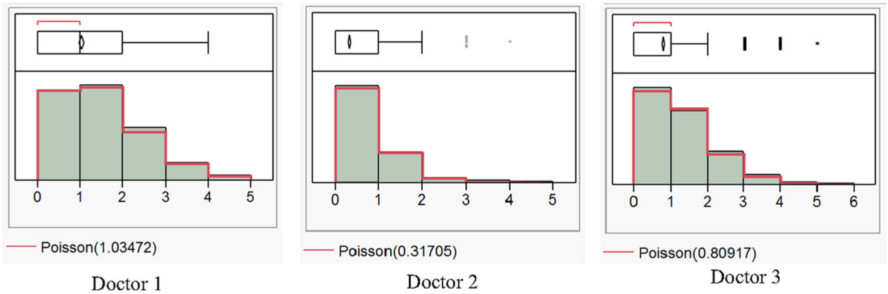

Fig. 3 Histograms of the patients served in each time slot

Table 2 Statistical results for service process

Doctor	Mean	SD	SE	χ^2	Prob > χ^2
SL	0.87	0.48	0.01	1229.28	0***
ZS	0.76	0.55	0.01	696.68	0***
QQ	0.97	0.22	0.01	1684.69	0***
GX	0.12	0.35	0.01	4.2	0.24
XW	0.88	0.51	0.02	409.19	0***
YY	1.03	0.97	0.03	5.62	0.34
YZ	0.87	0.56	0.02	277.33	0***
YC	0.8	0.57	0.01	526.41	0***
WX	0.87	0.57	0.01	1037.93	0***
LC	0.92	0.54	0.02	658.64	0***
YM	0.32	0.58	0.02	3.08	0.69
TS	0.91	1.06	0.02	72.91	0***
YF	0.81	0.9	0.02	5.49	0.48
MQ	0.14	0.39	0	7.18	0.13
WH	0.84	0.92	0.03	9.09	0.17
JZ	0.66	0.82	0.02	9.98	0.08*

*0.5 $< p$-value ≤ 0.1;
***p-value < 0.01

(Kopach et al. 2007). The literature, e.g., Kaandorp and Koole (2007) and Hassin and Mendel (2008), also use exponentially distributed service times in their scheduling models. Thus we propose the following hypothesis:

$\boldsymbol{H_0}$: the counts of departures in a slot follow Poisson distribution.
$\boldsymbol{H_1}$: the counts of departures in a slot do not follow Poisson distribution.

Table 2 shows the statistical results. Among the 16 data subsets, the counts of served patients in a slot follow Poisson distribution (p-value > 0.1) in 7 subsets. It is hard to claim that the service time follows exponential distribution according to our statistical results. The reason may be that the physicians can always adjust the service time according to the current workload, which results in endogeneity. But exponential service time is still a good approximation, concluding from literature and our data analysis.

Based on the above statistical analysis, we make following assumptions.

Assumption 3.1 General patients and complicated patients arrive in Poisson processes with arrival rate λ_G and λ_C respectively.

Assumption 3.2 Service time is exponentially distributed with mean $\frac{1}{\mu}$.

Although the statistical analysis indicates that sometimes the service time may not be exponentially distributed, assuming exponential distribution of service times is acceptable and a good approximation. In Sect. 5.6, we show that our main results still hold when the service times are deterministic.

4 The model and solution approach

4.1 The model

With Assumptions 3.1 and 3.2 on hand, MDP can be a suitable solution approach for the online stochastic scheduling problem described in Sect. 3.1. MDP is widely used to solve problems where sequential decisions are taken in each period with consideration of their future impacts. Major components of a MDP include decision epochs, states, actions, transition probabilities, and rewards (or costs) (Puterman 1990). We now describe these 5 key elements and explain how to solve the scheduling problem via MDP.

- *Decision epoch*: a decision is made whenever a general patient P_G comes for service. Since the patients arrive in a Poisson process, the decision epochs do not occur at fixed time points, meaning that the MDP model is a continuous-time MDP.
- *States*: states (denoted by $\boldsymbol{s} \in \mathcal{S}$) in the MDP model are the numbers of patients in the system. Let n_J (n_S) be the number of patients being served by and waiting for physician D_J (D_S), then $\boldsymbol{s} = (n_J, n_S)$. The bound of $n_J (n_S)$ is predetermined as the capacity of the system, denoted by $\overline{n}_J(\overline{n}_S)$, thus the set of states $\mathcal{S} = \{(n_J, n_S)|0 \leq n_J \leq \overline{n}_J,\ 0 \leq n_S \leq \overline{n}_S\}$.
- *Actions*: at each decision epoch, an action can be assigning a normal patient P_G to D_S or not. Let $\boldsymbol{a} = \{a(\boldsymbol{s}|\boldsymbol{s} \in \mathcal{S})\}$ denote a stochastic policy which maps each state to a distribution of actions, i.e., $a(\boldsymbol{s})$ is the probability of assigning P_G to D_S at state $\boldsymbol{s}$. Let $\mathcal{A}$ denote the set of all feasible policies, i.e., $\mathcal{A} = [0, 1]^{|\mathcal{S}|}$.
- *Transition probability*: whenever a patient joins the system or departs after the service, the state $\boldsymbol{s}$ changes. The transit of the states depends on which event occurs. By Assumptions 3.1 and 3.2, the time interval between two events is exponentially distributed, giving the model the Markovian Property. Let $p(\boldsymbol{s}_i, \boldsymbol{s}_j|a(\boldsymbol{s}_i))$ be the probability that the next state is $\boldsymbol{s}_j$ given the current state $\boldsymbol{s}_i$. Denote the transition matrix under policy $\boldsymbol{a}$ as $\boldsymbol{P}(\boldsymbol{a}) = \{p(\boldsymbol{s}_i, \boldsymbol{s}_j|a(\boldsymbol{s}_i))$ for all $\boldsymbol{s}_i, \boldsymbol{s}_j \in \mathcal{S}\}$. Table 3 shows transition probabilities.
- *Costs*: the cost associated with a state $\boldsymbol{s} = (n_G, n_S)$ consists of two parts. One is the waiting cost of the patients. The number of the patients waiting for services is $(n_G - 1)^+ + (n_S - 1)^+$, and they wait until the state changes. Let $\tau(\boldsymbol{s}|\boldsymbol{a})$ be the expected time interval from state $\boldsymbol{s}$ to next state under policy $\boldsymbol{a}$. Then $((n_G - 1)^+ + (n_S - 1)^+) \times \tau(\boldsymbol{s}|\boldsymbol{a})$ is the expected waiting time at state $\boldsymbol{s}$. The other one is the penalty cost due to deferment of the patients. At the boundary states, upcoming

Table 3 $p(s_i, s_j|\boldsymbol{a})$

s_i	s_j				
(n_G, n_S)	(n_G-1, n_S)	(n_G, n_S-1)	(n_G+1, n_S)	(n_G, n_S+1)	(n_G, n_S)
$0<n_G<\overline{n}_G$ $0<n_S<\overline{n}_S$	$\frac{\mu}{\lambda_G+\lambda_C+2\mu}$	$\frac{\mu}{\lambda_G+\lambda_C+2\mu}$	$\frac{(1-a(s_i))\lambda_G}{\lambda_G+\lambda_C+2\mu}$	$\frac{a(s_i)\lambda_G+\lambda_C}{\lambda_G+\lambda_C+2\mu}$	N.A.
$n_G=0$ $0<n_S<\overline{n}_S$	N.A.	$\frac{\mu}{\lambda_G+\lambda_C+\mu}$	$\frac{(1-a(s_i))\lambda_G}{\lambda_G+\lambda_C+\mu}$	$\frac{a(s_i)\lambda_G+\lambda_C}{\lambda_G+\lambda_C+\mu}$	N.A.
$0<n_G<\overline{n}_G$ $n_S=0$	$\frac{\mu}{\lambda_G+\lambda_C+\mu}$	N.A.	$\frac{(1-a(s_i))\lambda_G}{\lambda_G+\lambda_C+\mu}$	$\frac{a(s_i)\lambda_G+\lambda_C}{\lambda_G+\lambda_C+\mu}$	N.A.
$n_G=\overline{n}_G$ $0<n_S<\overline{n}_S$	$\frac{\mu}{\lambda_G+\lambda_C+2\mu}$	$\frac{\mu}{\lambda_G+\lambda_C+2\mu}$	N.A.	$\frac{a(s_i)\lambda_G+\lambda_C}{\lambda_G+\lambda_C+2\mu}$	$\frac{(1-a(s_i))\lambda_G}{\lambda_G+\lambda_C+2\mu}$
$0<n_G<\overline{n}_G$ $n_S=\overline{n}_S$	$\frac{\mu}{\lambda_G+\lambda_C+2\mu}$	$\frac{\mu}{\lambda_G+\lambda_C+2\mu}$	$\frac{(1-a(s_i))\lambda_G}{\lambda_G+\lambda_C+2\mu}$	N.A.	$\frac{a(s_i)\lambda_G+\lambda_C}{\lambda_G+\lambda_C+2\mu}$
$n_G=0$ $n_S=0$	N.A.	N.A.	$\frac{(1-a(s_i))\lambda_G}{\lambda_G+\lambda_C}$	$\frac{a(s_i)\lambda_G+\lambda_C}{\lambda_G+\lambda_C}$	N.A.
$n_G=\overline{n}_G$ $n_S=\overline{n}_S$	$\frac{\mu}{\lambda_G+\lambda_C+2\mu}$	$\frac{\mu}{\lambda_G+\lambda_C+2\mu}$	N.A.	N.A.	$\frac{\lambda_G+\lambda_C}{\lambda_G+\lambda_C+2\mu}$
$n_G=\overline{n}_G$ $n_S=0$	$\frac{\mu}{\lambda_G+\lambda_C+\mu}$	N.A.	N.A.	$\frac{a(s_i)\lambda_G+\lambda_C}{\lambda_G+\lambda_C+\mu}$	$\frac{(1-a(s_i))\lambda_G}{\lambda_G+\lambda_C+\mu}$
$n_G=0$ $n_S=\overline{n}_S$	N.A.	$\frac{\mu}{\lambda_G+\lambda_C+\mu}$	$\frac{(1-a(s_i))\lambda_G}{\lambda_G+\lambda_C+\mu}$	N.A.	$\frac{a(s_i)\lambda_G+\lambda_C}{\lambda_G+\lambda_C+\mu}$

patients may be deferred to the next day according to the policy. Let $\rho(\boldsymbol{s}|\boldsymbol{a})$ denote the probability of defering a patient at state $\boldsymbol{s}$ under policy $\boldsymbol{a}$. Then the total cost of state $\boldsymbol{s}$ under policy $\boldsymbol{a}$ is $f(\boldsymbol{s}|\boldsymbol{a}) = ((n_G-1)^+ + (n_S-1)^+) \times \tau(\boldsymbol{s}|\boldsymbol{a}) + h(\rho(\boldsymbol{s}|\boldsymbol{a}))$ where the unit waiting cost is normalized to be 1 and $h(\cdot)$ is the deferment cost function. $\boldsymbol{f}(\boldsymbol{a})$ denotes the vector of $f(\boldsymbol{s}|\boldsymbol{a})$. Table 4 show $\tau(\boldsymbol{s}|\boldsymbol{a})$ and $\rho(\boldsymbol{s}|\boldsymbol{a})$ in different cases.

Let $\boldsymbol{\pi}(\boldsymbol{a}) = \{\pi(\boldsymbol{s}|\boldsymbol{a})$ for all $\boldsymbol{s} \in \mathcal{S}\}$ be the steady state probabilities under policy $\boldsymbol{a}$, which can be calculated by

$$\boldsymbol{\pi}(\boldsymbol{a})\boldsymbol{P}(\boldsymbol{a}) = \boldsymbol{\pi}(\boldsymbol{a}). \tag{1}$$

Then we can use $\boldsymbol{\pi}(\boldsymbol{a})\boldsymbol{f}(\boldsymbol{a})$ as the expected average cost $\eta(\boldsymbol{a})$ under policy $\boldsymbol{a}$ since the MDP is ergodic. The objective is to find a policy $\boldsymbol{a}^*$ to minimize the expected total cost $\eta(\boldsymbol{a}^*)$, i.e.,

$$\min_{\boldsymbol{a}\in\mathcal{A}} \boldsymbol{\pi}(\boldsymbol{a})\boldsymbol{f}(\boldsymbol{a}) \qquad \textbf{(P)}$$

$$s.t.\ \boldsymbol{\pi}(\boldsymbol{a})\boldsymbol{P}(\boldsymbol{a}) = \boldsymbol{\pi}(\boldsymbol{a})$$

Table 4 $\tau(s|\boldsymbol{a})$, $\rho(s|\boldsymbol{a})$

$\boldsymbol{s} = (n_G, n_S)$	$\tau(\boldsymbol{s}\|\boldsymbol{a})$	$\rho(\boldsymbol{s}\|\boldsymbol{a})$
$0 < n_G < \bar{n}_G$ $0 < n_S < \bar{n}_S$	$\frac{1}{\lambda_G+\lambda_C+2\mu}$	0
$n_G = 0$ $0 < n_S < \bar{n}_S$	$\frac{1}{\lambda_G+\lambda_C+\mu}$	0
$0 < n_G < \bar{n}_G$ $n_S = 0$	$\frac{1}{\lambda_G+\lambda_C+\mu}$	0
$n_G = \bar{n}_G$ $0 < n_S < \bar{n}_S$	$\frac{1}{\lambda_G+\lambda_C+2\mu}$	$\frac{(1-a(\boldsymbol{s}_i))\lambda_G}{\lambda_G+\lambda_C+\mu}$
$0 < n_G < \bar{n}_G$ $n_S = \bar{n}_S$	$\frac{1}{\lambda_G+\lambda_C+2\mu}$	$\frac{a(\boldsymbol{s}_i)\lambda_G+\lambda_C}{\lambda_G+\lambda_C+2\mu}$
$n_G = 0$ $n_S = 0$	$\frac{1}{\lambda_G+\lambda_C}$	0
$n_G = \bar{n}_G$ $n_S = \bar{n}_S$	$\frac{1}{\lambda_G+\lambda_C+2\mu}$	$\frac{\lambda_G+\lambda_C}{\lambda_G+\lambda_C+2\mu}$
$n_G = \bar{n}_G$ $n_S = 0$	$\frac{1}{\lambda_G+\lambda_C+\mu}$	$\frac{(1-a(\boldsymbol{s}_i))\lambda_G}{\lambda_G+\lambda_C+\mu}$
$n_G = 0$ $n_S = \bar{n}_S$	$\frac{1}{\lambda_G+\lambda_C+\mu}$	$\frac{a(\boldsymbol{s}_i)\lambda_G+\lambda_C}{\lambda_G+\lambda_C+\mu}$

4.2 Policy iteration algorithm

4.2.1 Preliminary

Before we describe the algorithm to find the optimal policy $\boldsymbol{a}^*$, we first present a few fundamental results from Cao (2007).

Denote the *Potential* of state $\boldsymbol{s}$ under policy $\boldsymbol{a}$ as $g(\boldsymbol{s}|\boldsymbol{a})$, and it is defined as:

$$g(\boldsymbol{s}|\boldsymbol{a}) = f(\boldsymbol{s}|\boldsymbol{a}) - \eta(\boldsymbol{a}) + \sum_{\boldsymbol{s}' \in \mathcal{S}} p(\boldsymbol{s}, \boldsymbol{s}'|a(\boldsymbol{s}))g(\boldsymbol{s}'|\boldsymbol{a}). \tag{2}$$

$g(\boldsymbol{s}|\boldsymbol{a})$ can be viewed as the long-term "potential" contribution of state $\boldsymbol{s}$ to the expected average cost. Specifically, $f(\boldsymbol{s}|\boldsymbol{a}) - \eta(\boldsymbol{a})$ is the one-step contribution at the current state $\boldsymbol{s}$; and $\sum_{\boldsymbol{s}' \in \mathcal{S}} p(\boldsymbol{s}, \boldsymbol{s}'|\boldsymbol{a})g(\boldsymbol{s}'|\boldsymbol{a})$ is the expected long-term "potential" contribution of the next state. The matrix form of Eq. (2) is known as the Poisson equation:

$$(\boldsymbol{I} - \boldsymbol{P}(\boldsymbol{a}))\boldsymbol{g}(\boldsymbol{a}) + \eta(\boldsymbol{a})\boldsymbol{e} = \boldsymbol{f}(\boldsymbol{a}), \tag{3}$$

where $\boldsymbol{I}$ is the identity matrix with the same dimension as $\boldsymbol{P}$, and $\boldsymbol{e}$ is the unit vector with the same dimension as $\boldsymbol{f}$.

Because *potentials* are relative values, we cannot get a unique $\boldsymbol{g}(\boldsymbol{a})$ from Eq. (3), it need to be normalized by setting $\boldsymbol{\pi}(\boldsymbol{a})\boldsymbol{g}(\boldsymbol{a}) = \eta(\boldsymbol{a})$. We will have

$$(\boldsymbol{I} - \boldsymbol{P}(\boldsymbol{a}) + \boldsymbol{e}\boldsymbol{\pi}(\boldsymbol{a}))\boldsymbol{g}(\boldsymbol{a}) = \boldsymbol{f}(\boldsymbol{a}). \tag{4}$$

Lemma 1 (Cao 2007) *If there are two different policies, say* $\boldsymbol{a}_1$ *and* $\boldsymbol{a}_2$*, the* Expected Average Cost Difference *of these two policies is*

$$\eta(\boldsymbol{a}_1) - \eta(\boldsymbol{a}_2) = \boldsymbol{\pi}(\boldsymbol{a}_1)((\boldsymbol{P}(\boldsymbol{a}_1) - \boldsymbol{P}(\boldsymbol{a}_2))\boldsymbol{g}(\boldsymbol{a}_2) + (\boldsymbol{f}(\boldsymbol{a}_1) - \boldsymbol{f}(\boldsymbol{a}_2))) \tag{5}$$

Lemma 2 (Cao 2007) *A policy* $\boldsymbol{a}^*$ *is optimal if and only if*

$$\eta(\boldsymbol{a}^*)\boldsymbol{e} + \boldsymbol{g}(\boldsymbol{a}^*) = \min_{\boldsymbol{a}\in\mathcal{A}}\{\boldsymbol{P}(\boldsymbol{a})\boldsymbol{g}(\boldsymbol{a}^*) + \boldsymbol{f}(\boldsymbol{a})\}. \tag{6}$$

Lemma 3 (Cao 2007) *A policy* $\boldsymbol{a}^*$ *is optimal if and only if for all* $\boldsymbol{a} \in \mathcal{A}$ *we have*

$$\boldsymbol{P}(\boldsymbol{a}^*)\boldsymbol{g}(\boldsymbol{a}^*) + \boldsymbol{f}(\boldsymbol{a}^*) \le \boldsymbol{P}(\boldsymbol{a})\boldsymbol{g}(\boldsymbol{a}^*) + \boldsymbol{f}(\boldsymbol{a}). \tag{7}$$

4.2.2 Algorithm

Based on the above lemmas, we design our *Policy Iteration Algorithm* (PI Algorithm) for Problem (**P**) as **PI Algorithm** below.

```
Input: MDP model
Output: an optimal policy a*
initialization: guess an initial policy a_0, set k = 0;
while optimality is not achieved do
  | 1. calculate transition probability P(a_k) under policy a_k;
  | 2. solve the corresponding steady state probabilities π(a_k) by Equation (1);
  | 3. obtain the potential g(a_k) by Poisson Equation (4);
  | 4. solve min_{a∈A}{P(a)g(a_k) + f(a)} component-wisely; solution is denoted by a_{k+1};
  | if a_{k+1} == a_k then
  |   | optimality achieved;
  |   | a* := a_{k+1};
  |   | return a*
  | else
  |   | k := k + 1;
  | end
end
```

Algorithm 1: PI Algorithm

Theorem 1 PI Algorithm *stops in a finite number of iterations, and its output is an optimal solution of problem (**P**).*

Proof It follows Lemma 3. □

Proposition 1 *If* $f(\boldsymbol{s}|\boldsymbol{a})$ *only depends on* $a(\boldsymbol{s})$ *and is a linear function of* $a(\boldsymbol{s})$*, the optimal policy is a deterministic policy, i.e.,* $a(\boldsymbol{s})^* \in \{0, 1\}$ *for all* $\boldsymbol{s} \in \mathcal{S}$.

Proof Proof: see A.1. □

Recall that $f(\boldsymbol{s}|\boldsymbol{a}) = ((n_G - 1)^+ + (n_S - 1)^+) \times \tau(\boldsymbol{s}|\boldsymbol{a}) + h(\rho(\boldsymbol{s}|\boldsymbol{a}))$, and both $\tau(\boldsymbol{s}|\boldsymbol{a})$ and $\rho(\boldsymbol{s}|\boldsymbol{a})$ are linear functions of $a(\boldsymbol{s})$. If $h(\cdot)$ is also a linear function, then Proposition 1 holds. Consequently, the search space of $a(\boldsymbol{s})^*$ is dramatically narrowed. For each $\boldsymbol{s}_i \in \mathcal{S}$, instead of solving $\min_{\boldsymbol{a}\in\mathcal{A}}\{\boldsymbol{P}(\boldsymbol{a})\boldsymbol{g}(\boldsymbol{a}_k) + \boldsymbol{f}(\boldsymbol{a})\}$ state-wisely, we only need to compare the values of $\sum_{\boldsymbol{s}_j\in\mathcal{S}} p(\boldsymbol{s}_i, \boldsymbol{s}_j|a(\boldsymbol{s}_i))g(\boldsymbol{s}_j|\boldsymbol{a}_k) + f(\boldsymbol{s}_i|a(\boldsymbol{s}_i))$ when $a(\boldsymbol{s}_i) = 1$ and $a(\boldsymbol{s}_i) = 0$, and update $a(\boldsymbol{s}_i)$ using the one with smaller cost.

Proposition 2 *If $f(\boldsymbol{s}|\boldsymbol{a})$ only depends on $a(\boldsymbol{s})$ and is a linear function of $a(\boldsymbol{s})$, then the optimal policy $\boldsymbol{a}^*$ satisfies that, if $a^*(n_G - 1, n_S) = 1$, $a^*(n_G + 1, n_S) = 1$, $a^*(n_G, n_S - 1) = 1$ and $a^*(n_G, n_S + 1) = 1$, then $a^*(n_G, n_S) = 1$; if $a^*(n_G - 1, n_S) = 0$, $a^*(n_G + 1, n_S) = 0$, $a^*(n_1, n_S - 1) = 0$ and $a^*(n_G, n_S + 1) = 0$, then $a^*(n_G, n_S) = 0$.*

Proof Proof: see A.2. □

Proposition 2 indicates that in the optimal policy, if the action at the neighbors of state $\boldsymbol{s} = (n_G, n_S)$ is 1, then the action at that state is 1; and if in the optimal policy the action at the neighbors of state $\boldsymbol{s} = (n_G, n_S)$ is 0, then the action at that state is 0. In each iteration of the *PI Algorithm*, this property allows us to obtain a possible policy without solving the minimization problem for all states.

5 Simulation and policy evaluation

Recall that all theoretical results in Sect. 4 are based on the stationary analysis. However, the decision horizon is not infinite in reality thus the policy obtained by *PI Algorithm* may not always be optimal in practices. To assess the value of the scheduling policy, we use simulation to compare it with two policies which are widely used in Chinese hospitals.

The first policy is to let patients choose the physicians by themselves, called *"Free to Choose Policy"*. It is widely used in the hospitals without scheduling center. Since patients always prefer senior physicians as long as they are available, this policy can be mathematically described as follows.

$$a(\boldsymbol{s}) = \begin{cases} 1 & \text{if} \quad n_S < \overline{n}_S \\ 0 & \text{if} \quad n_S = \overline{n}_S. \end{cases} \qquad \text{(Free to Choose Policy)}$$

The second one is the *"Myopic Policy"*. In some hospitals, the patients are assigned by staffs or information systems. However, without careful design, hospitals only look at the current status and assign the patients to the shortest queue (Ge 2016). This policy can be mathematically described as follows.

$$a(\boldsymbol{s}) = \begin{cases} 1 & \text{if} \quad n_S < n_J \\ 0 & \text{if} \quad n_S \geq n_J. \end{cases} \qquad \text{(Myopic Policy)}$$

We simulate the patients arriving and departing processes under Free to Choose Policy, Myopic Policy and the policy obtained by the *PI Algorithm* (call it *MDP Policy*). Fairness is guaranteed by using the same seeds of random number generator. The simulation horizon is 8 h. During these 8 h, we record the waiting time of each patient and count the number of served patients as well as the deferred patients. We use the average waiting time per served patient and the number of deferred patients as the evaluation criteria. To eliminate the impact of variability, for each parameter setting, we conduct the simulation 100 times and take the average values. The range of arrival rates for both types of patients is from 0.0001 to 0.0028 per second with 0.0003 as the step size. The range of service rate is from 0.0007 to 0.0034 with the same step size. We use linear function with the coefficient ranging from 5400 to 21,600 to represent the deferment cost $h(\cdot)$. The coefficient is the unit deferment cost. For example, coefficient 5400 means that the cost of deferring one patient is as same as letting one patient wait for 5400 s.

5.1 Impact of changes in λ_C

Figure 4 shows the average waiting time under these three policies when complicated patients' arrival rate λ_C changes. Generally we can see that when λ_C is higher, patients in the system wait longer. We can also find that Free to Choose Policy always has the worst performance. However, when λ_C increases, the disadvantage of Free to Choose Policy becomes larger firstly then slackens. The insights here is that, when λ_C is small, Myopic Policy and MDP Policy assign a proper number of normal patients to D_S, while Free to Choose Policy assigns much more, hence Free to Choose Policy performs more poorly. When there are more complicate patients, it becomes harder to rely on Myopic Policy and MDP Policy to improve the system. The difference between Myopic Policy and MDP Policy is not significant especially when λ_C is low. However, we can still see that MDP Policy is better than Myopic Policy when λ_C is relatively high. The reason is that the MDP Policy assigns less patients to D_S anticipating that there may be many complicated patients in the future.

Figure 5 shows how the number of deferred patients changes with λ_C under these three policies. We can see that when λ_c is higher, and the system has more patients deferred. The performance of Free to Choose Policy is poor especially when λ_C is large. Myopic Policy and MDP Policy have very close numbers of deferred patients when λ_C is small, while Myopic Policy has more patients deferred when λ_C is large. The insight here is that, when λ_C is small, there is seldom a long queue in the system, then all policies have few deferred patients. When λ_C gets large, MDP Policy assigns a right number of general patients to D_S, thus there are the fewest deferred patients with MDP Policy.

5.2 Impact of changes in λ_G

Things are different when we look at the arrival rate of normal patients λ_G. Figure 6 shows the impacts on the average waiting time. With increasing λ_G, the waiting time of patients in system using Myopic Policy and MDP Policy decreases

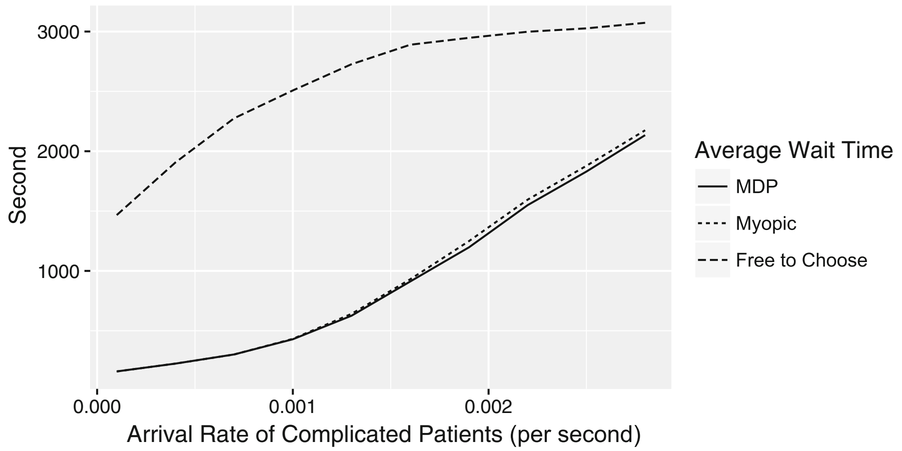

Fig. 4 Impact of changes of λ_C on the average waiting time, $\lambda_G = 0.0019$, $\mu = 0.0022$, deferment cost: 10,800

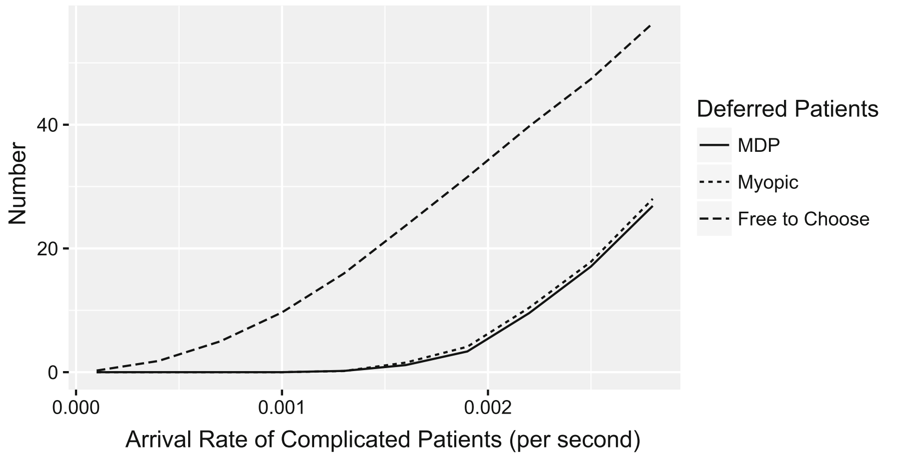

Fig. 5 Impact of λ_C on the deferred patients, $\lambda_G = 0.0019$, $\mu = 0.0022$, deferment cost: 10,800

firstly, then increases again. When λ_G is small, the normal patients seldom wait, thus most of the waiting cost is resulted from the complicated patients. With λ_G increasing, the total waiting time does not increase a lot, while more patients are served. That is why we have shorter average waiting time per patient when λ_G increases a little bit. We can also see that the disadvantage of Free to Choose Policy is more significant and MDP Policy outperforms Myopic Policy more when λ_G is higher.

Figure 7 shows the changes of the number of deferred patients with changing λ_G. Free to Choose Policy results in the most deferred patients. Under this policy, normal patients always go to D_S as long as D_S is available, thus the more normal patients, the more deferred patients. For Myopic Policy and MDP Policies, the number of deferred patients slightly increases when λ_G is small, then it increases

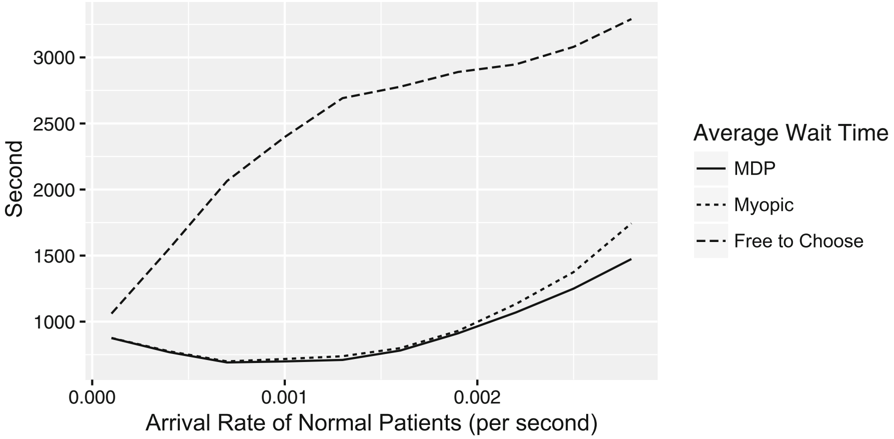

Fig. 6 Impact of changes of λ_G on the average waiting time, $\lambda_C = 0.0016$, $\mu = 0.0022$, deferment cost: 10,800

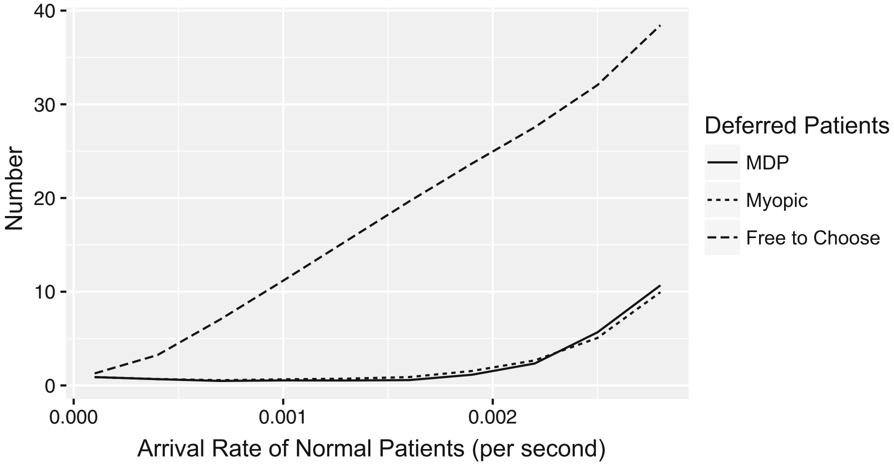

Fig. 7 Impact of changes of λ_G on the deferred patients, $\lambda_C = 0.0016$, $\mu = 0.0022$, deferment cost: 10,800

rapidly when λ_G goes large. The reason is that, when λ_G is not too large, these two policies can balance the workloads by assigning some normal patients to D_S. However the effects weaken when λ_G is large since the system is already overloaded.

5.3 Impact of changes in μ

Figure 8 shows the average waiting time with different service rates. Generally we can see that when μ increases, average waiting time under all policies decreases. Free to Choose Policy still has the longest average waiting time. Comparing Myopic Policy

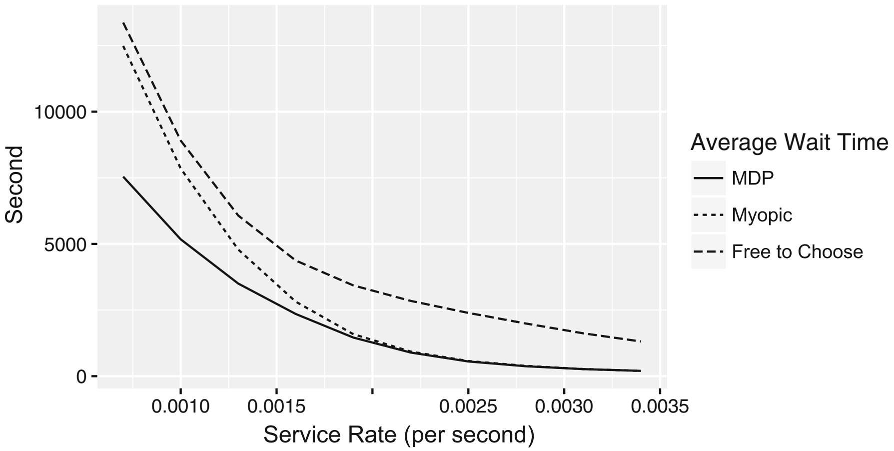

Fig. 8 Impact of changes of μ on the average waiting time $\lambda_G = 0.0019$, $\lambda_C = 0.0016$, deferment cost: 10,800

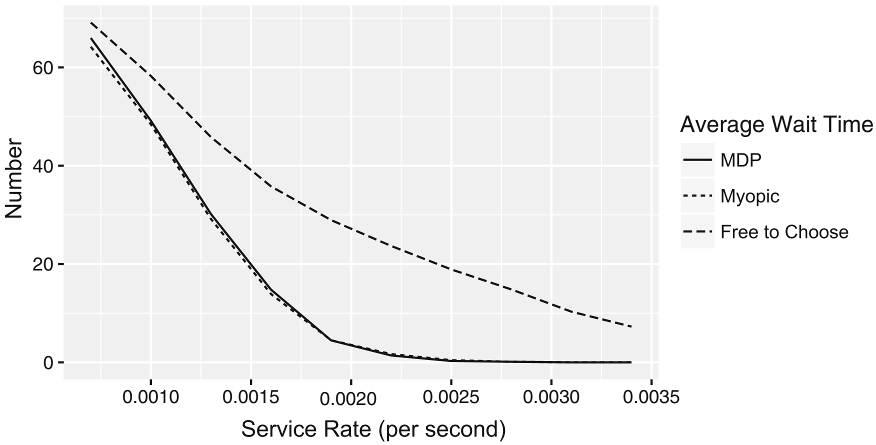

Fig. 9 Impact of changes of μ on the deferred patients, $\lambda_G = 0.0019$, $\lambda_C = 0.0016$, deferment cost: 10,800

and MDP Policy, MDP Policy outperforms Myopic Policy a lot when μ is small, while large μ diminishes its advantage. The reason is that both policies cannot make a big improvement when the system is not busy.

Changes in the number of deferred patients are shown in Fig. 9. It decreases in the service rate μ under all policies. Compared with the other two policies, the Free to Choose Policy increases first, then decreases when service rate increases. The insight here is that when service rate is too high, there are few patients to defer no matter how normal patients are assigned; when service rate is too low, the system is highly congested hence all policies do not help much.

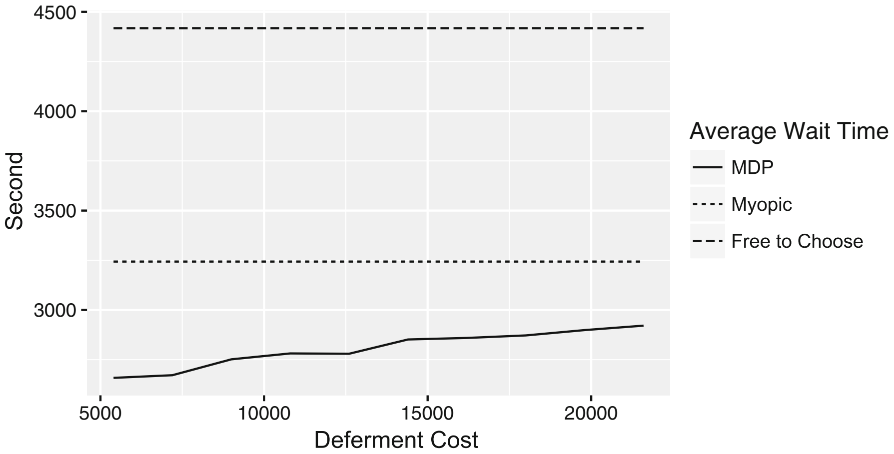

Fig. 10 Impact of changes of deferment cost on the average waiting time, $\lambda_G = 0.0019$, $\lambda_C = 0.0019$, $\mu = 0.0016$

5.4 Impact of deferment cost

This section examines the performances of three different policies with different deferment costs under different scenerios. We consider two scenarios: one is a busy system, in which we set $\lambda_G = \lambda_C = 0.0019$, $\mu = 0.0016$; the other one is a light system, in which we set $\lambda_G = \lambda_C = 0.0019$, $\mu = 0.0022$.

When the system is busy, Figs. 10 and 11 show the impacts of deferment cost on the average waiting time and the number of deferred patients, respectively. With deferment cost increasing, the waiting time of the patients under MDP Policy increases, while it stays stable under Free to Choose Policy and Myopic Policy. It is straightforward that Free to Choose Policy and Myopic Policy do not take the deferment cost into account, hence the number of deferred patients does not change, and neither does with the average waiting time. However, the deferment cost has big influences on MDP Policy. When it is larger, MDP Policy defers less patients and the average waiting time increases due to more served patients with larger deferment cost. From Fig. 10, we can see that the average waiting time under MDP Policy is much shorter compared with the other two policies, even with large deferment cost.

From Figs. 12 and 13, we can see that the general trends of the average waiting time and the number of deferred patients in a light system are the same as a busy system. However, the performances of Myopic Policy and MDP Policy are very close when system is underloaded. These results indicate that Myopic Policy has a good performance in a light system, and MDP Policy loses its competitiveness.

5.5 Comparison of Myopic Policy and MDP policy

As discussed before, although we can see much improvement via MDP Policy, compared with Myopic Policy in terms of the average waiting time, we cannot claim MDP

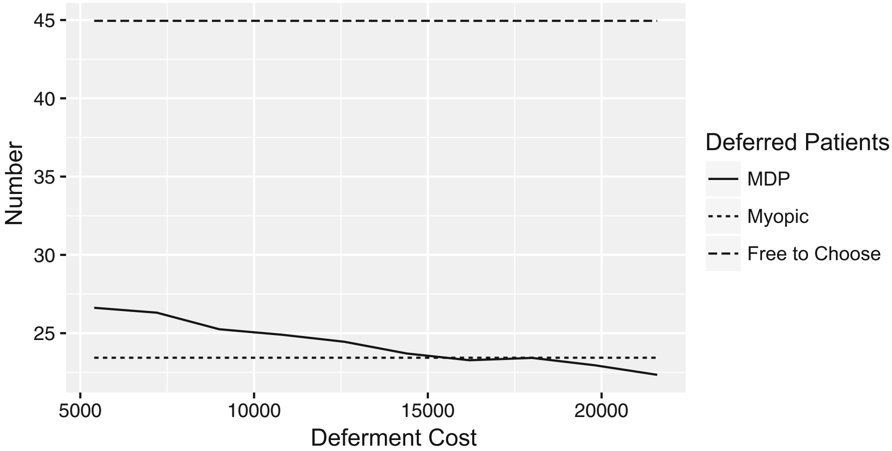

Fig. 11 Impact of changes of deferment cost on the deferred patients, $\lambda_G = 0.0019$, $\lambda_C = 0.0019$, $\mu = 0.0016$

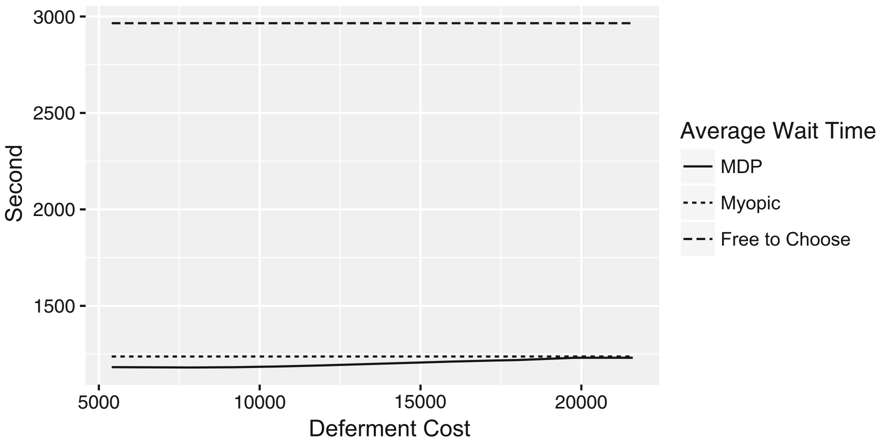

Fig. 12 Impact of changes of deferment cost on the average waiting time, $\lambda_G = 0.0019$, $\lambda_C = 0.0019$, $\mu = 0.0022$

Policy is always better because it results in slightly more deferred patients in some cases. In this section, we compare Myopic Policy and MDP Policy in details. To systematically show their performance differences, we conduct the paired tests on the experimental results. Since there is no impact of deferment cost under Myopic Policy, we only include the samples with different λ_G, λ_C or μ. The sample size is 30. What we are interested in is whether the differences between the average waiting time, the number of deferred patients and the total cost are same or not. The total cost is calculated by (the average waiting time $\times$ # of served patients + 10800 $\times$ # of deferred patients), and the tested difference is calculated by (the value of MDP Policy $-$ the value of Myopic Policy). The hypotheses to be tested are summarized as follows.

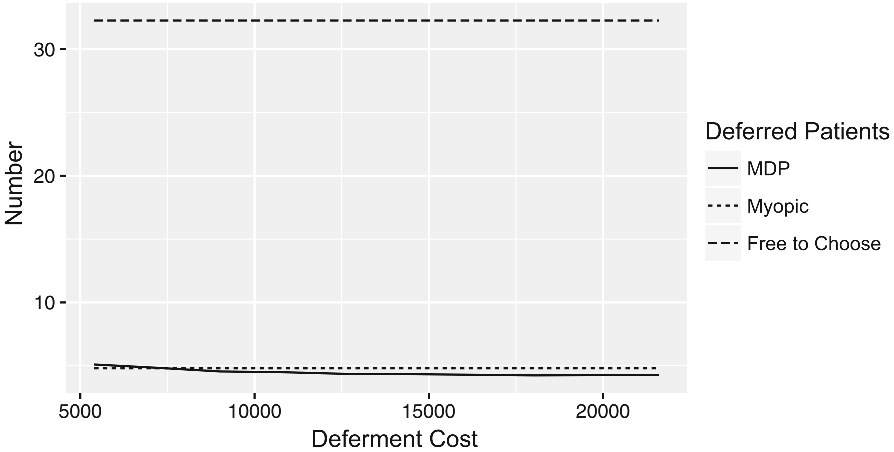

Fig. 13 Impact of changes of deferment cost on the number of deferred patients, $\lambda_G = 0.0019$, $\lambda_C = 0.0019$, $\mu = 0.0022$

Table 5 Statistical results of the paired tests

Test item	H0	H1	Mean	SD	p-value
Average waiting time	$= 0$	< 0	-344.87	1017.3	0.0735*
# of deferred patients	$= 0$	< 0	-0.0033	0.5977	0.9758
Total cost	$= 0$	< 0	$-19{,}638$	43192.8	0.0187**

*0.5 $< p$-value ≤ 0.1; **0.1 $< p$-value ≤ 0.05

$\boldsymbol{H_0^A}$: the difference of the average waiting time equals to 0.
$\boldsymbol{H_1^A}$: the difference of the average waiting time is smaller than 0.
$\boldsymbol{H_0^B}$: the difference of the number of deferred patients equals to 0.
$\boldsymbol{H_1^B}$: the difference of the number of deferred patients is smaller than 0.
$\boldsymbol{H_0^C}$: the difference of the total cost equals to 0.
$\boldsymbol{H_1^C}$: the difference of the total cost is smaller than 0.

Table 5 shows the statistical results of the paired tests. We can see that for the differences in average waiting time and the total cost, we can reject the Null Hypothesis and accept the Alternative Hypothesis, which means that the difference is smaller than 0. As for the number of deferred patients, we do not accept the alternative hypothesis, but we cannot reject the Null Hypothesis neither. Consequently, we claim that compared with Myopic Policy, the MDP Policy reduces the average waiting time per served patient, without increasing the number of deferred patients, thus it has lower total cost.

Table 6 Statistical results of paired tests with deterministic service times

Test item	H0	H1	Mean	SD	p-value
Average waiting time	$= 0$	< 0	− 296.02	940.39	0.0953*
# of deferred patients	$= 0$	< 0	− 0.6613	2.32	0.1292
Total cost	$= 0$	< 0	− 29,419	787,612	0.0762*

*$0.5 < p\text{-value} \leq 0.1$

5.6 Simulation in deterministic service times environment

Recall that in Sect. 3, the empirical results suggest that in some cases the service time is not exponentially distributed. In our model and the solution process, we ignore this fact for elegant analytical results. To fully evaluate the value of the MDP Policy, we implement Myopic Policy and MDP Policy with the deterministic service times. Since Free to Choose Policy has the worst performance with the exponential service times, we do not consider it here. We run 100 times of simulations in 30 different parameter settings, and do the paired tests to compare the performance differences between Myopic Policy and MDP Policy in terms of the average waiting time per served patient, the number of deferred patients and the total cost. The total cost, the differences and the hypotheses to be tested are the same as in Sect. 5.5. Table 6 shows the statistical results of the paired tests. It has similar results as in Table 5, meaning that even with the deterministic service times, the MDP Policy outperforms Myopic Policy in terms of the average waiting time per served patient and the total cost, without increasing the number of deferred patients. Hence we claim that MDP Policy has better performance than Myopic Policy even when service times are deterministic.

5.7 Managerial implications

We have asked three questions in Sect. 1. Based on the theoretical analysis and simulation results, we have the following answers and managerial insights for practitioners.

First, it is worth spending money on setting a scheduling center. Our analysis shows that Free to Choose Policy always has the worst performance, indicating that the lack of a scheduling center results in high cost and low efficiency in term of both patients waiting time and the number of deferred patients. A trivial policy like Myopic Policy can significantly improve the whole process in all cases. We believe that the saved cost by a simple scheduling policy far exceeds the cost of setting up a scheduling center.

Second, when the system is not busy or the service time is relatively short, the resource (available time of physicians) is relatively sufficient to serve all patients, thus the performances of Myopic Policy and MDP Policy are very close. However, it takes time and money to develop new information systems and train the staffs when a complex policy is adopted, while its value is not significant, hence it is suitable to implement “Myopic Policy” in this case.

However, health care resources are insufficient especially in the developing countries. According to the interview with Doctor Ge (2016) as well as the data set obtained from Shanghai General Hospital, the outpatient department is always busy and heavily loaded. Based on our analysis, we anticipate that the performance of Myopic Policy is far away from the one of MDP Policy when the system is busy. it means that the scheduling policy need to be carefully designed to achieve good performance of the whole system. MDP Policy is theoretically the optimal policy in steady state, and shows a big advantage under various scenarios in our simulation study.

6 Extension and conclusions

6.1 Extension

In this paper, we study the scheduling policy design problem for the case of two physicians. Now we show that the MDP solution approach can be easily extended to the case with more than one physician in each group. All analytic results remain valid with minor adjustment. In this case, the state can be denoted as:

- $\boldsymbol{s} = \{n_{J1}, n_{J2} \ldots, n_{Ji} \ldots, n_{S1}, n_{S2} \ldots, n_{Si} \ldots\}$
  - $n_{Ji}(\boldsymbol{s})$ is the number of patients being served by or waiting for ith physician in junior group at state $\boldsymbol{s}$.
  - $n_{Si}(\boldsymbol{s})$ is the number of patients being served by or waiting for ith physician in senior group at state $\boldsymbol{s}$.
  - $w(\boldsymbol{s})$ is the total number of working physician at state $\boldsymbol{s}$, i.e., the counts of $n_{xi} > 0$ for $x \in J, S$ and all i.

For the state $\boldsymbol{s}$ in the interior, the transition probability from $\boldsymbol{s}$ to another state $\boldsymbol{s}'$ is:

- if for some $x \in J, S$ and some i, $n_k(\boldsymbol{s}') = n_k(\boldsymbol{s}) - 1$ for $k = xi$ and $n_k(\boldsymbol{s}') = n_k(\boldsymbol{s})$ for $k \neq xi$,

$$p(\boldsymbol{s}, \boldsymbol{s}'|\boldsymbol{a}) = \frac{\mu}{\lambda_C + \lambda_G + w(\boldsymbol{s})\mu},$$

- if for $Ji = \arg\min\{n_{J1}, n_{J2} \ldots, n_{Ji} \ldots\}$, $n_k(\boldsymbol{s}') = n_k(\boldsymbol{s}) + 1$ for $k = Ji$ and $n_k(\boldsymbol{s}') = n_k(\boldsymbol{s})$ for $k \neq Ji$,

$$p(\boldsymbol{s}, \boldsymbol{s}'|\boldsymbol{a}) = \frac{(1 - a(\boldsymbol{s}))\lambda_G}{\lambda_C + \lambda_G + w(\boldsymbol{s})\mu},$$

- if for $Si = \arg\min\{n_{S1}, n_{S2} \ldots, n_{Si} \ldots\}$, $n_k(\boldsymbol{s}') = n_k(\boldsymbol{s}) + 1$ for $k = Si$ and $n_k(\boldsymbol{s}') = n_k(\boldsymbol{s})$ for $k \neq Si$,

$$p(\boldsymbol{s}, \boldsymbol{s}'|\boldsymbol{a}) = \frac{a(\boldsymbol{s})\lambda_G + \lambda_C}{\lambda_C + \lambda_G + w(\boldsymbol{s})\mu},$$

- for other $\boldsymbol{s}'$,

$$p(\boldsymbol{s}, \boldsymbol{s}'|\boldsymbol{a}) = 0.$$

The expected time interval from $\boldsymbol{s}$ to another state $\boldsymbol{s}'$ is:

$$\frac{1}{\lambda_C + \lambda_G + w(\boldsymbol{s})\mu}.$$

For the states at the boundary, $p(\boldsymbol{s}, \boldsymbol{s}'|\boldsymbol{a})$, $\tau(\boldsymbol{s}|\boldsymbol{a})$ and $\rho(\boldsymbol{s}|\boldsymbol{a})$ can be obtained in the same way as in Sect. 4. We omit them here. Then $\boldsymbol{P}(\boldsymbol{a})$ and $\boldsymbol{f}(\boldsymbol{a})$ can be formulated and we can use the *PI Algorithm* in Sect. 4.2.2 to obtain the optimal policy.

6.2 Conclusions

How to design the online scheduling policy with the heterogeneous customers and service providers has long been a problem faced by the practitioners in many service sectors. This problem is more serious in health care industry. We study the process of outpatient clinic services in China and find that there is a big problem in outpatient scheduling process. The trade off lies in how to schedule the patients to minimize the waiting time and the deferment penalty simultaneously, taking into consideration of the heterogeneity of both the patients and the physicians as well as the uncertainty in arrival and service processes. We model the problem as an online stochastic scheduling problem. From the literature review and the empirical results from teh data analysis, we assume the Poisson arrivals and exponential service times. Instead of solving it by traditional machine scheduling theory and simulation, we take advantage of Markov decision process (MDP) and propose a new algorithm for this scheduling problem. First, we show how to obtain an MDP Policy by solving this problem optimally in steady state using Policy Iteration Algorithm. Wwe then show the structural properties of MDP Policy to reduce the complexity of Policy Iteration Algorithm. Second, we design the numerical experiments to compare the MDP Policy with two policies (Free to Choose Policy and Myopic Policy) used in practices by simulating various scenarios. The numerical results show that the MDP Policy has the best performance across all scenarios, especially when the system is heavily loaded. "Myopic Policy" is acceptable compared with Free to Choose Policy. Its performance is quite close to MDP Policy when the system is not busy. For extensions, we conduct simulations with deterministic service times to demonstrate the robustness of our model. In addition, we also show how to extend our model to multi-server case.

To summarize, this paper has three contributions. First, we describe the whole process of patient clinic service in China, and model it as an online stochastic scheduling problem. We answer the question whether the hospital does need the distribution/scheduling center or not through the performance comparisons among three different policies. Second, we develop a new scheduling policy based on Markov decision process, and show that this policy outperforms "Myopic Policy" and "Free to Choose Policy", which are widely used in practice, by extensive numerical experiments over a fairly broad set of parameter settings. Third, by applying Markov decision process, we can deal with the scheduling problem with stochastic arrival times and processing time more conveniently. This Markov decision process can be extended to other scheduling problems with different application areas.

There are several ways to extend the current research. First, the model with more types of patients and physicians can be explored. Second, through empirical data analysis, we show in most cases, the arrival process is Poisson and the service times follow exponential distribution. However, models with general distributions for the arrival processes and the service times are worthy studying.

Acknowledgements The authors are grateful to Shanghai General Hospital for providing data and help with this research. The authors are listed alphabetically and they contribute equally to this work.

Appendix: Proofs

A.1 Proof of Proposition 1

If $h(\cdot)$ is a linear function, then in the minimization problem of each iteration, the objective function in is also a linear function of $a(\boldsymbol{s})$, then the optimal solution must be achieved among extreme points, which is either 1 or 0.

A.2 Proof of Proposition 2

If $f(\boldsymbol{s}|\boldsymbol{a})$ only depends on $a(\boldsymbol{s})$ and is a linear function of $a(\boldsymbol{s})$, and under policy $\boldsymbol{a}$, $a(n_G - 1, n_S) = 1$, $a(n_G + 1, n_S) = 1$, $a(n_G, n_S - 1) = 1$ and $a(n_G, n_S + 1) = 1$, then we have, $g((n_G, n_S)|\boldsymbol{a}) \geq g((n_G - 1, n_S + 1)|\boldsymbol{a})$, $g((n_G + 2, n_S)|\boldsymbol{a}) \geq g((n_G + 1, n_S + 1)|\boldsymbol{a})$, $g((n_G + 1, n_S - 1)|\boldsymbol{a}) \geq g((n_G, n_S)|\boldsymbol{a})$ and $g((n_G + 1, n_S + 1)|\boldsymbol{a}) \geq g((n_G, n_S + 2)|\boldsymbol{a})$. Since $g((n_G + 1, n_S)|\boldsymbol{a}) = \frac{1-p_1-p_2}{2}(g((n_G, n_S)|\boldsymbol{a}) + g((n_G + 1, n_S - 1)|\boldsymbol{a})) + p_1 \min(g((n_G + 2, n_S)|\boldsymbol{a}), g((n_G + 1, n_S + 1)|\boldsymbol{a})) + p_2 g((n_G + 1, n_S + 1)|\boldsymbol{a})$ and $g((n_G, n_S + 1)|\boldsymbol{a}) = \frac{1-p_1-p_2}{2}(g((n_G, n_S)|\boldsymbol{a}) + g((n_G - 1, n_S + 1)|\boldsymbol{a})) + p_1 \min(g((n_G, n_S + 2)|\boldsymbol{a}), g((n_G + 1, n_S + 1)|\boldsymbol{a})) + p_2 g((n_G, n_S + 2)|\boldsymbol{a})$, where $p_1 = \frac{\lambda_G}{\lambda_G+\lambda_C+2\mu}$ and $p_2 = \frac{\lambda_C}{\lambda_G+\lambda_C+2\mu}$, thus we have $g((n_G + 1, n_S)|\boldsymbol{a}) \geq g((n_G, n_S + 1)|\boldsymbol{a})$, which means that $a(n_G, n_S) = 1$.

If $f(\boldsymbol{s}|\boldsymbol{a})$ only depends on $a(\boldsymbol{s})$ and is a linear function of $a(\boldsymbol{s})$, and under policy $\boldsymbol{a}$, $a(n_G - 1, n_S) = 0$, $a(n_G + 1, n_S) = 0$, $a(n_G, n_S - 1) = 0$ and $a(n_G, n_S + 1) = 0$, then we have, $g((n_G, n_S)|\boldsymbol{a}) \leq g((n_G - 1, n_S + 1)|\boldsymbol{a})$, $g((n_G + 2, n_S)|\boldsymbol{a}) \leq g((n_G + 1, n_S + 1)|\boldsymbol{a})$, $g((n_G + 1, n_S - 1)|\boldsymbol{a}) \leq g((n_G, n_S)|\boldsymbol{a})$ and $g((n_G + 1, n_S + 1)|\boldsymbol{a}) \leq g((n_G, n_S + 2)|\boldsymbol{a})$. Since $g((n_G + 1, n_S)|\boldsymbol{a}) = \frac{1-p_1-p_2}{2}(g((n_G, n_S)|\boldsymbol{a}) + g((n_G + 1, n_S - 1)|\boldsymbol{a})) + p_1 \min(g((n_G + 2, n_S)|\boldsymbol{a}), g((n_G + 1, n_S + 1)|\boldsymbol{a})) + p_2 g((n_G + 1, n_S + 1)|\boldsymbol{a})$ and $g((n_G, n_S + 1)|\boldsymbol{a}) = \frac{1-p_1-p_2}{2}(g((n_G, n_S)|\boldsymbol{a}) + g((n_G - 1, n_S + 1)|\boldsymbol{a})) + p_1 \min(g((n_G, n_S + 2)|\boldsymbol{a}), g((n_G + 1, n_S + 1)|\boldsymbol{a})) + p_2 g((n_G, n_S + 2)|\boldsymbol{a})$, where $p_1 = \frac{\lambda_G}{\lambda_G+\lambda_C+2\mu}$ and $p_2 = \frac{\lambda_C}{\lambda_G+\lambda_C+2\mu}$, thus we have $g((n_G + 1, n_S)|\boldsymbol{a}) \leq g((n_G, n_S + 1)|\boldsymbol{a})$, which means that $a(n_G, n_S) = 0$.

References

Ahn JH, Hornberger JC (1996) Involving patients in the cadaveric kidney transplant allocation process: a decision-theoretic perspective. Manag Sci 42(5):629–641

Begen MA, Queyranne M (2011) Appointment scheduling with discrete random durations. Math Operat Res 36(2):240–257

Bennett CC, Hauser K (2013) Artificial intelligence framework for simulating clinical decision-making: a markov decision process approach. Artif Intell Med 57(1):9–19

Breg BP, Denton BP (2017) Fast approximation methods for online scheduling of outpatient procedure centers. INFORMS J Comput 29(4):631–644

Cao XR (2007) Stochastic learning and optimization: a sensitivity-based approach. Springer, Berlin

Cayirli T, Veral E (2003) Outpatient scheduling in health care: a review of literature. Prod Oper Manag 12(4):519–549

Cayirli T, Veral E, Rosen H (2006) Designing appointment scheduling systems for ambulatory care services. Health Care Manag Sci 9(1):47–58

Cayirli T, Yang KK, Quek SA (2012) A universal appointment rule in the presence of no-shows and walk-ins. Prod Oper Manag 21(4):682–697

Chen G, Shen ZJM (2007) Probabilistic asymptotic analysis of stochastic online scheduling problems. IIE Trans 39(5):525–538

Chou MC, Liu H, Queyranne M, Simchi-Levi D (2006) On the asymptotic optimality of a simple on-line algorithm for the stochastic single-machine weighted completion time problem and its extensions. Oper Res 54(3):464–474

Fetter RB, Thompson JD (1966) Patients' waiting time and doctors' idle time in the outpatient setting. Health Serv Res 1(1):66

Ge X (2016) Personal communication, October 14, 2016, Shanghai General Hospital, Shanghai

Green LV, Savin S, Wang B (2006) Managing patient service in a diagnostic medical facility. Oper Res 54(1):11–25

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

Hassin R, Mendel S (2008) Scheduling arrivals to queues: a single-server model with no-shows. Manag Sci 54(3):565–572

Hu C, Lovejoy WS, Shafer SL (1996) Comparison of some suboptimal control policies in medical drug therapy. Oper Res 44(5):696–709

Kaandorp GC, Koole G (2007) Optimal outpatient appointment scheduling. Health Care Manag Sci 10(3):217–229

Kim SH, Whitt W (2014) Are call center and hospital arrivals well modeled by nonhomogeneous Poisson processes? Manuf Serv Oper Manag 16(3):464–480

Knowledge@Wharton (2013) ticking time bombs: Chinas health care system faces issues of access, quality and cost. http://knowledge.wharton.upenn.edu/article/ticking-time-bombs-chinas-health-care-system-faces-issues-of-access-quality-and-cost/

Kolesar P (1970) A Markovian model for hospital admission scheduling. Manag Sci 16(6):B-384

Kong Q, Lee CY, Teo CP, Zheng Z (2013) Scheduling arrivals to a stochastic service delivery system using copositive cones. Oper Res 61(3):711–726

Kopach R, DeLaurentis PC, Lawley M, Muthuraman K, Ozsen L, Rardin R, Wan H, Intrevado P, Qu X, Willis D (2007) Effects of clinical characteristics on successful open access scheduling. Health Care Manag Sci 10(2):111–124

LaGanga LR, Lawrence SR (2012) Appointment overbooking in health care clinics to improve patient service and clinic performance. Prod Oper Manag 21(5):874–888

Li J, Dong M, Ren Y, Yin K (2015) How patient compliance impacts the recommendations for colorectal cancer screening. J Comb Optim 30(4):920–937

Magni P, Quaglini S, Marchetti M, Barosi G (2000) Deciding when to intervene: a markov decision process approach. Int J Med Inform 60(3):237–253

Megow N, Vredeveld T (2006) Approximation in preemptive stochastic online scheduling. Springer, Berlin

Megow N, Uetz M, Vredeveld T (2006) Models and algorithms for stochastic online scheduling. Math Oper Res 31(3):513–525

Patrick J (2012) A markov decision model for determining optimal outpatient scheduling. Health Care Manag Sci 15(2):91–102

Puterman ML (1990) Markov decision processes. Handb Oper Res Manag Sci 2:331–434

Qu X, Peng Y, Shi J, LaGanga L (2015) An MDP model for walk-in patient admission management in primary care clinics. Int J Prod Econ 168:303–320

Rising EJ, Baron R, Averill B (1973) A systems analysis of a university-health-service outpatient clinic. Oper Res 21(5):1030–1047
Schaefer AJ, Bailey MD, Shechter SM, Roberts MS (2005) Modeling medical treatment using Markov decision processes. In: Operations research and health care. Springer, pp 593–612
Schulz AS (2008) Stochastic online scheduling revisited. In: Combinatorial Optimization and Applications. Springer, pp 448–457
Sevcik KC (1974) Scheduling for minimum total loss using service time distributions. J ACM (JACM) 21(1):66–75
Sloan TW (2007) Safety-cost trade-offs in medical device reuse: a Markov decision process model. Health Care Manag Sci 10(1):81–93
Swartzman G (1970) The patient arrival process in hospitals: statistical analysis. Health Serv Res 5(4):320–9
Tan Z, Zhang A (2013) Online and semi-online scheduling. Springer, New York
Wang S, Liu N, Wan G (2016) Managing appointment-based services in the presence of walk-in customers. In: Working paper. Columbia University, New York
Weiss G (1995) On almost optimal priority rules for preemptive scheduling of stochastic jobs on parallel machines. Adv Appl Probab 27:821–839
Wiesche L, Schacht M, Werners B (2016) Strategies for interday appointment scheduling in primary care. Health Care Manag Sci. https://doi.org/10.1007/s10729-016-9361-7
Zacharias C, Pinedo M (2014) Appointment scheduling with no-shows and overbooking. Prod Oper Manag 23(5):788–801