Dynamic Surgery Management under Uncertainty

E. Gökalp[a], N. Gülpınar[b*], X. V. Doan[b]

[a] *School of Management, University of Bath, Bath, BA2 7AY, UK.*

[b] *Warwick Business School, The University of Warwick, Coventry, CV4 7AL, UK.*

Abstract

Real-time surgery management involves a complex and dynamic decision-making process. The duration of surgeries cannot be known until the actual surgery is completed. Furthermore, disruptions like an equipment failure or the arrival of a non-elective surgery can occur simultaneously. Thus, the assignment of surgeries needs to be updated, as disruptions occur, to minimize their effects. In this paper, we present a stochastic dynamic programming approach to the surgery allocation problem with multiple operating rooms under uncertainty. Given an elective list for the day, the dynamic optimization model minimizes the number of surgeries uncompleted at the end of the shift and the total waiting times of patients during the day weighted with respect to their urgency level. Due to the curse of dimensionality, we apply an approximate dynamic programming algorithm to solve the stochastic dynamic surgery management model. Computational experiments are designed to demonstrate the performance of the proposed algorithm and its applicability in practical settings. The results show that the approximate dynamic programming algorithm provides a good approximation to the optimum policy and leads to some managerial insights.

Keywords: Reactive scheduling, uncertainty modelling, surgery management, approximate dynamic programming.

1. Introduction

Operating rooms are seen as engines of hospitals since they are one of the most profitable healthcare services (Carey et al. 2011). According to HFMA (2005), surgeries generate forty percent of the overall revenue in UK hospitals. On the other hand, the operating rooms account for the majority of hospitals' operational capacity (Macario et al. 1995), as they consume a significant amount of physical resources such as beds and equipments as well as human resources with different levels of expertise. Given its impact on the revenue and resource requirements, surgery management is one of the most crucial tasks for healthcare

*Corresponding author

Email addresses: elvan.gokalp@warwick.ac.uk (E. Gökalp[a]), Nalan.Gulpinar@wbs.ac.uk (N. Gülpınar[b]), Vinh.Doan@wbs.ac.uk (X. V. Doan[b])

professionals involving complex and dynamic decision-making stages. According to reports of the UK Department of Health (2016), existing management practices are not able to achieve performance targets, such as average time to receive a service, and thus need to be supported by analytical decision-making tools. For a review on different decision-making stages in surgery management, readers are referred to Erdogan et al. (2011).

An operating room serves both elective and non-elective patients and is subject to various disruptions. The surgery of an elective patient is scheduled days or weeks ahead whereas a non-elective patient needs to be served immediately or within a short period of time. We use (elective and non-elective) patient and surgery interchangeably in this paper. Non-elective surgeries create disruptions to elective schedules. Besides, the uncertainty in surgery durations can lead to some uncompleted surgeries at the end of the shift that need to be rescheduled. These are called last-minute cancellations. The surgery cancellations not only create discontent among patients, they also impact future schedules (Schofield et al. 2005). According to the National Health Service in England (NHS England, 2021), 1.1% of all elective surgeries (around 79,470 in total) were cancelled by NHS hospitals due to nonclinical reasons between April 2019 and March 2020.

Capacity shortage was reported as the main reason for these cancellations (Campbell and Arnett, 2015). As Fregene et al. (2017) mention, the waiting time for surgery is seen as another contributor of patient dissatisfaction during the surgery day and the long waiting times also negatively affect the quality of care. As these findings illustrate, hospital managers are under pressure to develop practical and effective strategies for the management of operating rooms in order for the overall patient dissatisfaction as well as operational costs to be reduced.

Real-time surgery management, so-called reactive surgery scheduling, involves a decision problem dealing with assignment of surgeries to operating rooms during a planning horizon (typically a day or a week), monitoring the surgeries as well as taking appropriate actions in case of delays or cancellations. The real-time management of operating rooms also includes updating schedules or rescheduling of surgeries whenever any disruption, such as staff unavailability or patient no-shows, occurs (for instance, see Stuart and Kozan, 2012). Moreover, uncertainties on arrival of non-elective patients and surgery durations play an important role in surgery scheduling practices and must be taken into account in the decision process (Van Riet and Demeulemeester, 2015). Given that the realisation time of disruptions as well as their potential effects in the future are not known precisely in advance, the real-time management of multiple operating rooms is further complicated by the dynamic and stochastic nature of the decision-making process.

In general, hospital managers develop different strategies to handle the possible impact of those disruptions. As reported by Van Riet and Demeulemeester (2015), two alternative strategies have been widely applied in practice. The first one treats non-elective patients in isolation from the scheduled surgeries by reserving dedicated room(s) based on the predicted demand. Due to estimation or forecasting errors, this strategy may result in an inefficient schedule that consequently leads to revenue loss because the operating room and medical staff remain idle during those specified time slot(s) if no non-elective patient arrives. The second strategy (which is referred to as an 'online approach') accommodates non-elective surgeries by allowing staff overtime and cancellation of electives in a reactive fashion; for instance, see Ozkarahan (2000), and Blake et al. (2002). Wullink et al. (2007) show that the online approach usually provides a better performance in terms of waiting time, staff overtime, and operating room utilisation. On the other hand, this causes patient discontent and extra cost due to cancellations and overtime (Hosseini, 2012). A hybrid approach, combining these two strategies, has also been used in several hospitals by introducing some buffers among elective schedules to accommodate potential non-elective surgeries (Van Riet and Demeulemeester, 2015). There are not many studies in real-time surgery management dealing with a comprehensive analysis of all relevant uncertainties and their impact on various performance criteria, which are crucial to produce rigorous surgery management strategies. Moreover, the evolving and dynamic nature of the real-time surgery management are overlooked in most of the studies. In this paper, we aim to address these issues and to find effective policies to manage multiple operating rooms such that patient satisfaction is maximized while overall operational costs are minimized. In particular, our contributions are summarized as follows:

- We present a stochastic dynamic programming approach to model the real-time surgery management problem over multiple operating rooms during a day. Unlike the related studies in the literature, our model takes into account uncertainties concerning non-elective arrivals and surgery durations and employs different performance criteria such as the waiting time of surgeries during the day and the number of surgeries that need to be rescheduled at the end of the day. Due to the large state space, finding exact solutions for the real-size instances is computationally intractable.
- We therefore propose an approximate dynamic programming (ADP) algorithm to tackle the curse of dimensionality. To the best of our knowledge, ADP has not been widely applied for solving the real-time surgery management problem. Silva and Souza (2020) use approximate dynamic programming with an integer programming model for surgical scheduling. This study differs from our paper from the modelling and solution perspectives that will be more elaborated in the literature review section. We design a series of computational experiments to illustrate performance of proposed approaches

and compare the performance of the ADP algorithm with a myopic heuristic. We investigate the impact of various elective surgery scheduling rules on the overall performance using real data. The computational results illustrate the practical applicability of the underlying optimization model with problem instances of 10 operating rooms, which can be solved within two hours. Moreover, they show that the proposed algorithm provides a good approximation to the optimal policy.

The remaining sections of this paper are organized as follows. Section 2 provides a brief review of the most related literature on real-time surgery management. In Section 3, we introduce a stochastic dynamic programming formulation of the daily surgery management problem that considers allocation of surgeries over multiple operating rooms under uncertain surgery durations and non-elective arrivals. Section 4 describes the ADP based solution approach in more detail. In Section 5, we explain the setting and data related issues for the computational experiments and also present our results to derive managerial insights using the ADP based policies. We summarize our findings and final remarks in Section 6.

2. A Brief Review of Related Work

The operational research community has conducted extensive studies on various strategic and tactical decision-making problems in surgery management. The reader is referred to Cardoen et al. (2010), Gur (2018), and Rahimi and Gandomi (2021) for a comprehensive review on strategic and tactical decisions using online and offline approaches. However, real-time surgery management has not received enough attention, as mentioned in Erdogan et al. (2011), Guerriero and Guido (2011), and Van Riet and Demeulemeester (2015). Although appointment management is closely related to surgery scheduling problems, distinguishing differences exist between these research areas in terms of the impact of inherent uncertainties and disruptions as well as the dynamic nature of surgical procedures (Gupta and Denton, 2008). The real time production scheduling problem for a manufacturing company has been widely studied by academics such as Billaut and Roubellat (1996), Wu et al. (1999), Megow et al. (2006), Mohring (2000), Aloulou and Portmann (2003). These studies differ from reactive surgery scheduling problems due to the nature of healthcare services, which involve patients as well as significant operational uncertainties such as surgery durations. Thus, the literature review includes only the surgery management studies that are the most relevant to our study.

Most studies in the surgery management literature focus on scheduling of elective surgeries (e.g., Freeman et al., 2016, Ozen et al., 2016; Zhang et al. 2019), or operating room staffing and scheduling (e.g., Bandi and Gupta, 2020). Several studies consider rescheduling of elective surgeries after a disruption to the initial schedule occurs. For instance, Spangenberg et al. (2018) have developed a mixed integer linear programming model to reschedule electives based on the information received such as completed surgeries and emergency arrivals. On the other hand, Xiao et al. (2018) present a stochastic knapsack model for the adaptive operating room scheduling problem to minimize overtime in view of a fixed sequence of next k patients. Their model ignores uncertainty on non-elective arrivals. Similarly, Spratt and Kozan (2019) introduce a real-time reactive framework for the surgical case sequencing problem, where the schedule of electives is reoptimized to maximize operating rooms' utilization in view of staff availability while non-elective surgeries are assigned to a separate operating room on arrival. However, they also ignore uncertain future non-elective arrivals. Jung et al. (2019) introduce heuristics to determine how much capacity to assign for elective patients such that any emergency surgery can take place. In cases of the arrival of emergency surgery, remaining electives are rescheduled under deterministic settings of known surgery durations and arrival of emergency surgeries in the future.

Among all these models developed for scheduling of electives, there are several studies taking into account various uncertainties. Hooshmand et al. (2018) use a genetic algorithm to solve the optimization model that schedules electives during a day under uncertain surgery duration. Non-elective surgeries are not considered and the rescheduling of electives is also allowed only at specific time periods during the day. A waiting list of elective patients and priority (urgency levels) of their surgeries are taken into account to reduce the waiting time and overtime of operating rooms by Zhang et al. (2019). They consider the uncertain arrivals of non-electives who have to be operated within the day, but they do not have different urgency levels attached. They develop a Markov decision model with dead ends for operating room planning in view of dynamic patient priority. These studies diverge from our paper because they focus on scheduling of elective patients. Astaraky et al. (2015) have developed a stochastic and dynamic programming model to schedule the elective surgery demand to available slots in advance of several days/weeks by taking into account the ward congestion. Due to the large problem size, they also propose an ADP approach to solve the model. Vandenberghe et al. (2021) introduce a mixed integer programming model to determine the optimum sequencing and room allocation of surgeries in a single day under uncertain surgery durations and emergency arrivals by minimizing waiting times of emergency patients and the number of cancellations. The emergency surgeries are assumed to be undertaken whenever an elective is finished in a room, thus delaying the elective schedule. They use a sample average approximation over possible scenarios and also propose a genetic algorithm. Similarly, Maghzi et al. (2019), developing a mixed integer linear programming model, take priorities of patients to schedule elective patients in advance. In addition, they include medical equipment and staff into the scheduling process. Multi-objective optimization is alternatively used by

Table 1: A classification of research papers on real-time surgery management

Research papers	Modelling			Decision		Uncertainty		Cost		Solution		Scheduling	
	IP	TS	MDP	C	A	PA	SD	Cc	Wc	Exact	Inexact	PS	RS
Batun (2011)		✓			✓	✓				✓		✓	
Hosseini (2012)			✓		✓	✓	✓				✓	✓	
Stuart & Kozan (2012)	✓			✓		✓	✓	✓		✓			✓
Erdem et al. (2012)	✓								✓		✓		✓
Van Essen et al. (2012)	✓				✓				✓		✓		✓
Zhang et al. (2014)		✓		✓			✓		✓	✓	✓	✓	
Duma & Aringhieri (2015)	Simulation				✓		✓	✓			✓		✓
Borgman (2017)	Simulation					✓	✓		✓		✓	✓	
Heydari & Saoudi (2016)		✓			✓	✓				✓		✓	
Silva & Souza (2020)			✓	✓	✓	✓	✓	✓	✓		✓	✓	
Kamran et al. (2020)	✓				✓	✓	✓	✓	✓		✓		✓
Wang et al. (2015)	✓				✓	✓					✓		✓
This paper			✓		✓	✓	✓	✓	✓	✓	✓		✓

IP: integer program, TS: two-stage stochastic program, MDP: Markov decision process
C: cancellation, A: assignment, PA: patient arrival, SD: surgery duration
Cc: cancellation cost, Wc: waiting time cost
PS: proactive scheduling, RS: reactive scheduling

Khalfalli et al. (2019), Guido et al. (2017) and Moreno et al. (2018).

Table 1 classifies the most relevant studies on the real-time surgery management in comparison to our paper in terms of modelling and solution approaches as well as relevant decisions, inherent uncertainties and different cost components. Different methods based on the reactive and proactive scheduling approaches are used to model real-time surgery decision-making problems. The reactive decision-making models for real-time surgery management have been developed by Stuart and Kozan (2012), van Essen et al. (2012), Duma and Aringhieri (2015) and Erdem et al. (2012). In particular, Stuart and Kozan (2012) focus on the reactive surgery scheduling problem with random non-elective arrivals and formulate the problem as a single machine scheduling problem with sequence dependent processing times and due dates by including the priorities of elective and non-elective surgeries. They consider two conflicting objectives to maximize the number of accepted non-elective patients and to minimize the number of cancelled elective surgeries. They assume that surgery durations are sequence dependent and follow independent log-normal distributions. The underlying optimization model is solved by the branch-and-bound algorithm which produces a list of surgeries that are expected to be late.

There are several studies dealing with the real-time surgery management problem under stochastic disruptions that affect the existing schedule of elective surgeries. For instance, Kamran et al. (2020) assign elective patients to time blocks when the initial schedule is disrupted. They propose a column generation method to solve the underlying mixed integer program. Miao and Wang (2021) also develop an optimization model assuming that surgery duration is deterministic. They propose to schedule patients with certain buffers and insert any emergency patient arrival between them. In case of any disruption to the original schedule during the day, they suggest rerunning the same model with modified objective function. Spratt and Kozan (2021) propose a similar approach by re-scheduling elective patients when a disruption occurs. Unlike these two studies, Wang et al. (2015) introduce a partial rescheduling where effects of inserting a random emergency patient are taken into consideration when generating elective surgeries' schedule offline.

Several authors have introduced inexact solution methods (including heuristics and metaheuristic approaches) due to computational difficulties of the reactive surgery scheduling problem. For instance, van Essen et al. (2012) model the reactive surgery scheduling problem as an integer linear program to minimize the deviances from stakeholders' preferences in order to determine a new elective surgery schedule in case of a disruption. They show that the reactive surgery scheduling problem for multiple rooms is NP-hard and provide constructive heuristics for real-sized instances. Similarly, Erdem et al. (2012) construct a deterministic mixed integer linear programming model that reschedules elective patients to the operation and post-anaesthesia clinical care units when a non-elective patient arrives. They assume that non-elective patients may be rejected or accepted and the objective function consists of the costs of postponing electives and rejecting non-electives. A genetic algorithm is applied to solve real-sized problem instances. Duma and Aringhieri (2015) propose a hybrid approach combining simulation and optimization for the real-time surgery assignment problem. The optimization module consists of a metaheuristic algorithm (based on a greedy construction of an initial solution) and a local search to improve the initial solution. The simulation allows to simulate uncertain surgery durations to be incorporated into the optimization module. Borgman (2017) also introduces a simulation approach to evaluate different non-elective surgery management strategies such as dedicating emergency rooms and combining electives and non-electives. Duma and Aringhieri (2018) introduce an online optimization approach for the real time surgery management. This first reschedules waiting surgeries when a non-elective patient arrives during the day and then assigns the required resources for each surgery. Since a reactive decision-making approach ignores the dynamic nature of the real-time surgery management problem, the proactive scheduling models have been introduced by taking into account inherent uncertainties and multiple decision stages to obtain a (proactive) surgery scheduling policy. For example, Zhang et al. (2014) are concerned with the dynamic assignment of elective surgeries to multi operating rooms under uncertain surgery durations. They develop a two-stage stochastic programming model which identifies the next surgery to assign to any operating room that is currently available. Heydari and Saoudi (2016) consider the random non-elective arrivals and develop a two-stage stochastic model that reschedules the surgeries upon the arrival of an emergency. On the other hand, they do not model the operational decisions such as cancellation of electives and rejection of non-electives. Conversely, they assume that non-elective surgeries are always accepted. In addition, Batun (2011) develops a two-stage stochastic (mixed integer) optimisation model for the surgery rescheduling problem and applies L-shaped decomposition and progressive hedging algorithms to solve the problem. The first-stage decision determines the number of operating rooms to open in a day. After the uncertainty is realised, the second-stage decisions on rescheduling of surgeries are taken. Instead of modelling real-time surgery management decisions for a short planning period (like a day), a longer planning horizon is considered by Addis et al. (2016). They apply a rolling-horizon method to find a scheduling policy for elective surgeries. Once the optimal schedule for several weeks is obtained by solving the integer linear programming model, the first week's strategy is implemented. They then re-optimise the schedule by adding new arrivals and cancellations from the previous week. The algorithm carries on in the same manner until the end of planning horizon.

The real-time surgery management problem requires a multi-stage decision-making process since uncertainties on the non-elective arrivals and surgery durations may be revealed throughout the day more than once. This basically justifies the use of stochastic dynamic programming as an appropriate modelling approach for the real-time surgery management problem. Although stochastic dynamic programming has been widely used as a modelling approach for different stages of surgery management such as case-mix planning and elective surgery scheduling (e.g., Min and Yih, 2014; Gerchak et al., 1996; Lamiri et al., 2008), its application for the real-time surgery management is limited considering the dynamic nature of real-time surgery management. Hosseini (2012) studies the dynamic real-time surgery management problem under both patient arrival and surgery duration uncertainties. A Markov decision process (MDP) model is developed by considering two separate queues with random service durations for the elective and non-elective surgeries. The state space consists of the length of queues and idleness of operating rooms while the action space involves an assignment of a patient from any queue to an operating room, as soon as an operating room becomes free. The dynamic model minimizes the system-wide, long-run average costs relating to patient allocations. Unlike Hosseini (2012), our model assumes that elective and non-elective patients are not assigned to a queue. Thus, the operating room manager needs to make a decision on assigning a waiting surgery to a room as soon as the room becomes available. This leads to a different definition of state and action spaces, as we explain in the next section.

Silva and Souza (2020) also study an approximate dynamic programming approach with integer programming model for the stochastic surgical scheduling problem over a day by considering surgery durations and emergency arrivals. Although this paper can be seen as the closest paper to our study in terms of the underlying stochastic dynamic programming model, there are still fundamental differences in the way state and action spaces are created and the solution approach. First of all, they assume that an emergency surgery can be rejected and a scheduled surgery can be cancelled or start out of its time window. The problem in each time period is modelled as an integer linear programming model that minimizes the overall cost. Secondly, they propose a policy iteration method to solve the approximate dynamic programming model. Unlike their model, we consider surgery priorities when selecting which patient to assign to the next available operating room and aim to minimize the total number of surgeries uncompleted and the waiting times of surgeries, both weighted with the respective urgency levels.

The contributions of our paper to literature are two-fold in terms of modelling and solution approaches. Firstly, we model the real-time surgery management problem over multiple operating rooms in the presence of random non-elective arrivals and uncertain surgery duration. We assume that all electives are ready at the beginning of the day, as practiced in some hospitals. The decision-maker can choose which surgery to be operated at the next available operating room among all the waiting electives assigned to that day and non-electives which arrive during the day. However, the medical urgency of surgeries needs to be taken into consideration. Therefore, we assign a priority to both elective and nonelective patients (as common practice in most hospitals) and combine their medical priorities and waiting times. The weighted waiting time of both elective and non-elective patients with priority is incorporated in only a few studies in the literature; for instance, see Van Riet and Demeulemeester (2015). We consider two conflicting objectives in order to maximize the number of completed surgeries at the end of the shift and also to minimize total waiting times of patients during the day weighted with respect to their urgency level. This model provides flexibility on defining feasible types of surgeries that can be operated in a room. Secondly, due to the curse of dimensionality, we apply an approximate dynamic programming algorithm to solve the stochastic dynamic surgery management model. We design computational experiments using real data. We introduce a myopic heuristic to compare the performance of the approximate dynamic program and investigate the impact of elective scheduling strategies on the overall cost. Our empirical study shows that the approximate dynamic programming algorithm provides a good approximation to the optimum policy.

3. Stochastic Dynamic Programming Model

In this section, we first describe model assumptions and then introduce notation and a stochastic dynamic programming formulation of the surgery allocation problem with multiple operating rooms in the presence of random non-elective arrivals and uncertain surgery duration. We consider real-time surgery management of multiple operating rooms during a typical working day. Each operating room is well equipped to serve certain types of surgeries such as orthopaedic and heart operations. At the beginning of the working day, a daily list of elective surgeries for the operating theatre is usually ready. On the basis of the elective scheduling strategy, some critical operational decisions need to be made throughout the day as surgeries are carried out, and non-elective patients in different health conditions arrive randomly to the hospital. There can be multiple non-elective arrivals during a specific time period. Moreover, as soon as a surgery is completed in an operating room (that is dependent on the stochastic surgery durations), the decision-maker must also determine which of the waiting patients, including non-electives, should be operated next in that room. Assigning a non-elective patient to an operating room may lead to postponing pre-scheduled surgeries. Delays in surgeries may result in deterioration in health conditions of patients especially for urgent cases, and also decrease patient satisfaction as well as resource utilisation. Due to the delays and unexpected arrival of non-electives, some surgeries may still be waiting at the end of the day shift. Depending on their urgency levels, these surgeries need to be rescheduled to other days or completed in the night shift. Both cases generate a significant cost to the hospital management.

The dynamic surgery management problem (that will be formulated in this section) seeks the optimum assignment decisions at each time period under each possible realization of uncertainties (i.e. non-elective arrivals and surgery durations) so that the total number of surgeries uncompleted and the waiting times of surgeries, both weighted with the respective urgency levels, are minimized. In this paper, we model the urgency levels of patients with a discrete category called *priority* that is included in the objective function. Therefore, an emergency patient with the highest priority is naturally assigned as the next patient to operate.

Model Assumptions: We assume that all elective patients arrive to the hospital in the beginning of the day, as the common practice in some UK hospitals (for instance, see Fregene et al., (2017)). Note that no-shows (i.e., elective patients not showing up on their surgery day), are already included in the setting since we only consider the elective list ready at the beginning of the day.

We also assume that none of the non-elective surgeries are rejected by the hospital. However, there is a probability that a non-elective cannot be completed within the day shift. Note that, due to the problem formulation, this is only possible when the non-elective does not have the highest priority level, i.e. an emergency patient. Finally, the patients, both elective and non-elective, are assumed to be able to wait until the end of the day shift.

Notation: We consider a planning horizon as a typical working day and discretized into T time slots, represented by $t = 1, \cdots, T$. A *decision epoch* is defined as the time point where a decision is made. The time between decision epochs $t-1$ and t is called as *time slot* t for $t = 1, \cdots, T$. We assume that an equal length Δ (e.g., 15 minutes) is used for all time periods. In particular, $t = 0$ denotes the beginning of the working day when the initial schedule of elective surgeries is prepared. Table 2 presents a description of notation used in the mathematical formulation of the surgery management problem.

Table 2: Description of notation

	Model parameters
i, j, r	indices represent patient (or surgery), surgery types and operating rooms, respectively
m, R	different types of surgeries and total number of operating rooms, respectively
$\mathcal{F}_r$	types of surgeries that can be conducted at operating room r
T	planning horizon discretised by T time periods represented by decision epochs $t = 1, \cdots, T$
n_e	total number of elective patients pre-scheduled for the day
	State variables and actions
p_i, τ_i, d_i	priority, type and duration of surgery i
n^t	total number of surgeries arrived at time periods $t = 0, 1 \cdots, t-1, t$
a_i	arrival time of surgery i (specifically $a_i = 0$ indicates arrival time of elective patients)
η^{t+1}	total number of non-elective patients arriving at $t+1$
$\check{p}_i^{t+1}$, $\check{\tau}_i^{t+1}$	priority and type of non-elective patient i arriving at $t+1$
ω_i^t	binary variable represents whether patient i is waiting (1) or in an operating room (0) at t
ι_r^t	binary variable represents whether room r accommodates a patient (1) or is empty (0) at t
f_r^t	binary variable takes 1 if a surgery in room r is finished at time t; and 0, otherwise
x_r^t	binary variable represents whether a patient is assigned to room r at t or not

We use indices i, j and r to denote patient (or surgery), surgery types and operating rooms, respectively. The surgeries are categorized based on their expected durations (i.e., not the medical discipline and urgency levels of patients). There are m types of surgeries, each of which is represented by a probability distribution $P(j)$ of its random surgery time, for $j = 1, \ldots, m$. Each patient i is assigned with a priority p_i, $p_i \in \{1, \ldots, p_{\max}\}$. There are R operating rooms, each of which can only perform surgeries of some types $s \in \mathcal{F}_r$, where $\mathcal{F}_r \subseteq \{1, \ldots, m\}$.

States: In view of the model assumptions (stated above), we define a system state $\mathbf{S}^t$ at epoch t in terms of information related to patients' surgeries and operating rooms' status as follows. At the beginning of each epoch t, let n^t denote the total number of patients who have arrived to the operating rooms in previous time slots. Clearly, for $t = 0$, we have $n^0 = n_e$ where n_e is the number of elective surgeries scheduled for that considered period. For each patient i for $i = 1, \ldots, n^t$ at time t, we have information on patient's priority and arrival time. Let p_i and a_i represent patient i's priority and arrival time, respectively. Note that $a_i = 0$ for all $i = 1, \ldots, n^0$.

As mentioned earlier, surgery duration is assumed to be uncertain. The surgery time of patient i is represented by a random variable d_i. This follows a distribution $P_d(\tau_i)$ where $\tau_i \in \{1, \ldots, m\}$ represents the type of surgery of patient i. In addition, we introduce a binary variable ω_i^t to represent the current position of patient i at time t. More precisely, $\omega_i^t = 1$ if patient i at time t is still waiting for surgery and $\omega_i^t = 0$ if the patient's surgery has started in one of the operating rooms. Note that, at the beginning of planning horizon ($t = 0$), all patients are in waiting status; that is $w_i^0 = 1$ for $i = 1, \ldots, n^0$.

Information related to the current condition of operating rooms needs to be gathered at the beginning of each epoch. For each operating room r, at the beginning of each epoch t, let $\iota_r^t \in \{1, \ldots, n^t\} \cup \{0\}$ represent the patient who is currently assigned to that operating room. In particular, if there is no patient in operating room r, then it becomes $\iota_r^t = 0$. Clearly, we have $\iota_r^0 = 0$ for all operating rooms $r = 1, \ldots, R$ at the beginning of planning horizon. In addition, the decision-maker needs to know whether the surgery assigned to each operating room r is finished or not. Let us define a binary variable f_r^t for each operating room r at time t such that it takes 1 ($f_r^t = 1$) if the current surgery is completed; and zero ($f_r^t = 0$) otherwise. We can compute it as follows:

$$f_r^t = \begin{cases} \mathbb{I}(d_{\iota_r^t} = t - a_{\iota_r^t} \,|\, d_{\iota_r^t} > t - a_{\iota_r^t} - 1), & \text{if } \iota_r^t \neq 0, \\ 1, & \text{otherwise.} \end{cases}$$

where $\mathbb{I}(*)$ is the indicator function of '$*$'. Note that at the beginning of planning horizon, $f_r^0 = 1$ for all operating rooms r where $r = 1, \ldots, R$. By combining all information available about patients and operating rooms, a state of the system at decision epoch t is denoted as $\mathbf{S}^t = (n^t, \tau, \mathbf{p}, \mathbf{a}, \omega^t; \iota^t, \mathbf{f}^t)$.

Actions: Given a state $\mathbf{S}^t = (n^t, \tau, \mathbf{p}, \mathbf{a}, \omega^t; \iota^t, \mathbf{f}^t)$ involving information related to patients and operating rooms at time t, the decision-maker needs to assign patients to the operating rooms that become available. Let x_r^t denote the patient to be assigned to operating room r at decision epoch t. Thus, if room r is currently available (i.e., $f_r^t = 0$), then the patient with high priority will be assigned to that room; in other words, $x_r^t = \iota_r^t$. Then, the feasible set of x_r^t can be stated as follows;

$$\mathcal{X}_r^t = \{i \,:\, 1 \leq i \leq n^t \,|\, \omega_i^t = 1,\, p_i \in \mathcal{F}_r\} \cup \{0\}.$$

On the other hand, if no patient is assigned, then we have $x_r^t = 0$.

Update in State Space: Together with these decisions $\mathbf{x}^t$ made at time t, one can update the state of patients and operating rooms after the arrival of non-elective patients in the next time slot $t+1$, i.e., before the next epoch $t+1$, using the existing state $\mathbf{S}^t$. Let η^{t+1} be the total number of non-elective patients who arrive in time slot $t+1$. Each non-elective patient i has priority $\check{p}_i^{t+1}$ and requires a surgery of type $\check{\tau}_i^{t+1}$ for all $i = 1, \ldots, \eta^{t+1}$. The current status of patients and operating rooms is updated accordingly so that the new state $\mathbf{S}^{t+1} = (n^{t+1}, \check{\tau}, \check{\mathbf{p}}, \check{\mathbf{a}}, \omega^{t+1}; \iota^{t+1}, \mathbf{f}^{t+1})$ at time $t+1$ is constructed from state $\mathbf{S}^t$ at time t as follows:

$$
\begin{aligned}
n^{t+1} &= n^t + \eta^{t+1}, && \\
p_{n^t+i} &= \check{p}_i^{t+1}, && \forall\, i = 1, \ldots, \eta^{t+1}, \\
\tau_{n^t+i} &= \check{\tau}_i^{t+1}, && \forall\, i = 1, \ldots, \eta^{t+1}, \\
a_{n^t+i} &= t+1, && \forall\, i = 1, \ldots, \eta^{t+1}, \\
\omega_{x_r^t}^{t+1} &= 0, && \forall\, r = 1, \ldots, R,\ x_r^t \neq 0, \\
\omega_{n^t+i}^{t+1} &= 1, && \forall\, i = 1, \ldots, \eta^{t+1}, \\
\iota_r^{t+1} &= x_r^t, && \forall\, r = 1, \ldots, R.
\end{aligned}
$$

Multiple Objectives: We consider two conflicting objectives. One is to maximize the number of patients whose surgeries are completed in the day shift or, equivalently, to minimize dissatisfaction due to patients who have to be rescheduled or operated in the night shift. Together with patient priority, we formulate this objective as follows:

$$
c_d(\mathbf{S}^T) = \sum_{i=1}^{n^T} p_i \omega_i^T. \tag{1}
$$

This basically aims to eliminate the total number of patients, who have not been operated by the end of day, and potential discomfort among those patients with high emergency conditions. In addition, it is important to consider the second objective of minimizing the total waiting time at each time period, which can be written as follows:

$$
c_w^t(\mathbf{S}^t) = \sum_{i=1}^{n^t} p_i (t - a_i) \omega_i^t, \quad \forall\, t = 1, \ldots, T. \tag{2}
$$

We follow the weighted sum method of multi-objective optimization problems (e.g., see Allmendinger et al., (2017)) by setting a parameter $\alpha \geq 0$ to combine these two objectives into a single objective $C(\mathbf{S}^t)$ as follows:

$$
C(\mathbf{S}^t) = \alpha \cdot c_d(\mathbf{S}^T) + (1-\alpha) \sum_{t=1}^{T} c_w^t(\mathbf{S}^t). \tag{3}
$$

The parameter $0 \leq \alpha \leq 1$ reflects how important the first objective of minimizing the dissatisfaction of patients whose surgeries need to be rescheduled is with respect to the secondary objective of minimizing the waiting times. Note that, when $\alpha = 0$, only the secondary objective is considered, which means patients' rescheduling is not taken into account. On the other hand, for $\alpha = 1$, the first objective is optimized.

Dynamic Optimization Model: We are now ready to formulate the real-time surgery management problem as a dynamic programming model. Let $V_t(\mathbf{S}^t)$ denote the optimal objective value of the subsequent time periods after $\mathbf{x}^t$ has been decided given state $\mathbf{S}^t$ of patients and operating rooms.

As mentioned earlier, at decision epoch $t < T$, we need to make decisions $x_r^t \in \mathcal{X}_r^t$ for operating rooms $r = 1, \ldots, R$. The total waiting time at epoch $t+1$ only depends on the first n^t patients, not those arriving in that time period. However, state $\mathbf{S}^{t+1}$ is random because the number η^{t+1} of patients who arrive at time $t+1$ is random and it follows a fixed distribution denoted by P_a. In addition, for each arriving patient i, the surgery type ($\check{p}_i$) and priority ($\check{\tau}_i$), are also random and they follow independent discrete distributions (represented by P_p and P_τ, respectively). Accordingly, we can state them as $\check{p}_i \sim P_p$ and $\check{\tau}_i \sim P_\tau$.

Finally, the indicators $\mathbf{f}^{t+1}$ depend on the joint distribution of random surgery times of R patients indexed by x_r^t for $r = 1, \ldots, R$. Under the assumption that all these random variables are independent, the value function $V_t(\mathbf{S}^t)$ at state $\mathbf{S}^t$ can then be expressed as follows:

$$V_t(\mathbf{S}^t) \quad = \min_{x_r^t \in \mathcal{X}_r^t,\, r=1,\cdots,R} \Bigg\{ \alpha \cdot \sum_{i=1}^{n^t} \tau_i(t+1-a_i)\ \omega_i^{t+1}(\omega^t, \mathbf{x}^t) + \mathbb{E}_{\eta^{t+1}\sim P_a}\left[\mathbb{E}_{(\check{\mathbf{p}},\check{\tau})\sim(P_p\times P_\tau)^{\eta^{t+1}}}\left[\mathbb{E}_{\tau_{x_r^t}\sim P_d(\tau_{x_r^t}),\, r=1,\cdots,R}\left[V_{t+1}(\mathbf{S}^{t+1}(\mathbf{S}^t, \mathbf{x}^t))\right]\right]\right]\Bigg\}. \tag{4}$$

On the other hand, at decision epoch T, no decision needs to be made and we have:

$$V_T(\mathbf{S}^T) = c_d(\mathbf{S}^T). \tag{5}$$

The problem can then be written in a compact form as $\min_{\mathbf{x}} V_0(\mathbf{S}^0)$. The optimum policy, consisting of the optimum actions at each possible state, can be found by using the traditional backward value iteration technique (for instance, see Boyan and Littman, 2000).

As in most real-life stochastic dynamic programming applications, solving the real-time surgery allocation model is computationally expensive due to the large number of states. For example, the state space of a small-size problem instance, involving management of a single operating room over four time periods by using two initial electives and two possible surgery types possesses 18,432 states. Furthermore, the state space under the same problem setting becomes around six times larger than this when the number of operating rooms is doubled. Similarly, the action space is also large at the initial epochs where many waiting surgeries are to be assigned to an operating room. Due to the curse of dimensionality, the traditional backward dynamic programming algorithm is not suitable to solve the stochastic dynamic program

(4)-(5). Thus, we apply a simulation-based forward dynamic programming algorithm, as explained in the next section.

4. Approximate Dynamic Programming Approach

Approximate dynamic programming (ADP) is a simulation-based forward-looking dynamic programming method and has been widely applied for solving real-life stochastic dynamic programming models (Powel, 2009). It basically determines a strategy that steps forward through time starting from an initial state. Each sample path is generated using a Monte Carlo simulation from the same initial state. The value functions are evaluated for all states (in a look-up table) or updated at states on a random path (reached from the initial state) using aggregation of states or regression models.

In particular, for the real-time operating room planning problem under uncertainty, we consider an ADP approach with double-pass and a lookup table. In the double-pass algorithm, after all states in a sample path are visited, the algorithm goes backward from the last period until the initial state and adds values of states iteratively. Moreover, we implement a value-iteration method (unlike Silva and Souza, 2020 who applied a policy-iteration-based algorithm) since the underlying dynamic programming model involves a large state space and a comparatively small action set (e.g., see Sun and Li, 2013). A value-iteration based ADP algorithm updates state values at each iteration where as a policy-iteration-based algorithm improves the policy iteratively. Algorithm 1 displays the pseudo-code of the ADP algorithm with the value iteration, lookup table and double-pass approaches for the stochastic dynamic programming formulation of the surgery management problem presented in the previous section. Next, a brief summary of the main steps of Algorithm 1 follows.

An initial list of elective patients, their surgery types and priorities as well as the probability distributions of uncertain non-elective arrivals and surgery durations are supplied as inputs to the algorithm. The same initial state is used for all iterations of the algorithm. Each visited state and its corresponding approximate value are inserted into the lookup table. Let n denote the iteration counter and N be the maximum number of iterations set by the modeller. The value of initial state ($\mathbf{S}_1^0$) is chosen as zero. At each iteration n, a sample path for surgery types ($\check{\mathbf{p}}_n$) and priorities ($\check{\boldsymbol{\tau}}_n$) of non-elective arrivals are randomly generated from their probability distributions. In other words, random numbers of non-elective arrivals with different priorities and types (including the possibility of no arrival) are generated for each time period $t = 1, \cdots, T-1$ in the sample path. At each iteration n of Algorithm 1, we also simulate the surgery completions for continuing surgeries ($\mathbf{f}_n^t$) based on $\mathbf{S}_n^{t-1}$ and probability distributions of surgery

Algorithm 1: Pseudo code of the ADP algorithm

Step 0: Initialization
– Fix number of iterations N and parameter Γ. Initialize $n = 1$.
– Set the value function at initial state as $\overline{V}_1^0(\mathbf{S}_1^0) = 0$.
Step 1: Sample path generation
– Set $\overline{V}_t^n(\mathbf{S}_k^t) = \overline{V}_t^{n-1}(\mathbf{S}_k^t)$ for $k = 1, \cdots, n$ and $t = 1, \cdots, T$.
– Generate a sample path of $\check{\mathbf{p}}_n, \check{\boldsymbol{\tau}}_n$.
Step 2: State generation and evaluation
for $t = 1, \cdots, T-1$ **do**
– Generate $\mathbf{f}_n^t$ (based on $\mathbf{S}_n^{t-1}$) and a random number γ.
if $n \leq N/2$ and $\gamma \geq \Gamma$ **then**
– Randomly select $\mathbf{x_n^t}$ among the feasible action set $\mathcal{X}^t$.
else
– Find the greedy action $\mathbf{x_n^t}$ by solving (4) based on state values stored in the lookup table.
– If a future state doesn't exist in the lookup table, then $\overline{V}_{t+1}^n(\mathbf{S}_n^{t+1}) = 0$.
end if
– Compute $V_t^n(\mathbf{S}_n^t)$ using (4).
– Update state variables
end for
Step 3: Value function approximation
for $t = T-1, \cdots, 1$ **do**
– Compute $V_t^n(\mathbf{S}_n^t) = C(\mathbf{S}_n^t) + V_{t+1}^n(\mathbf{S}_n^{t+1})$.
if state $(\mathbf{S}_n^t)$ exists in the lookup table, **then**
– Compute $\overline{V}_t^n(\mathbf{S}_n^t) = (1 - \beta_{n-1})\overline{V}_t^{n-1}(\mathbf{S}_n^t) + \beta_{n-1} V_t^n(\mathbf{S}_n^t)$,
else
– Set $\overline{V}_t^n(\mathbf{S}_n^t) = V_t^n(\mathbf{S}_n^t)$.
end if
end for
Step 4. Update iteration number $n := n + 1$.
– If $n \leq N$, go to *Step 1*. Otherwise, go to *Step 5*.
Step 5. Return all value function approximations $(\overline{V}_t^N)$ for $t = 1, \cdots, T$.

durations for $t = 1, \cdots, T-1$. Let $V_t^n(*)$ define value function computed at iteration n given a state $(*)$.

For a generated state $\mathbf{S}_n^t$, Algorithm 1 finds a greedy action $x_n^t \in \mathcal{X}^t$ and computes the value function $V_t^n(\mathbf{S}_n^t)$ by using the optimality equation (4). The approximate values are stored in a lookup table. For any future state required in calculation of the greedy action that has not been visited before, its value is assumed to be 0. Then, state $\mathbf{S}_n^{t+1}$ is computed on the basis of the greedy action x_n^t, and parameters of state $\mathbf{S}_n^t$. Once all possible states in the planning horizon are visited, the algorithm goes backward in time and recursively adds values of all future states (in the sample path) into V_t^n for $t = T-1, \cdots, 1$.

Let $\overline{V}_t^n(\mathbf{S}_n^t)$ represent the approximated value function (that is stored in the lookup table) at state $\mathbf{S}_n^t$ for iteration $n = 1, \cdots, N$ at time $t = 1, \cdots, T$. If state $\mathbf{S}_n^t$ is visited first time, we add its value $V_t^n(\mathbf{S}_n^t)$ to the lookup table, i.e. $\overline{V}_t^n(\mathbf{S}_n^t) = V_t^n(\mathbf{S}_n^t)$. Otherwise, we calculate

$$\overline{V}_t^n(\mathbf{S}_n^t) = (1-\beta_{n-1})\overline{V}_t^{n-1}(\mathbf{S}_n^t) + \beta_{n-1}V_t^n(\mathbf{S}_n^t),$$

where β_n is defined as a smoothing parameter. Since state values are expected to approach their exact values through iterations, β_n is formulated as a parametric affine function of n (Powell, 2009). Finally, the values stored inthe lookup for all states visited until iteration n are carried over to the next iteration: that is $\overline{V}_t^{n+1}(\mathbf{S}_k^t) = \overline{V}_t^n(\mathbf{S}_k^t)$, for $k = 1, \cdots, N$ and $t = 1, \cdots, T$.

Note that the optimization problem may have multiple optima since different actions may result in the same state value. In order to increase the number of states to be visited, we employ different exploration strategies. For instance, we randomly select a greedy action in case of multiple optima. Specifically, we check whether a randomly generated number is lower than a fixed constant, represented as Γ. If so, then the greedy action is randomly selected among the feasible actions. Otherwise, it is chosen randomly among the optimum actions. However, this strategy may result in suboptimal policies and decreases the exploitation that is defined as the degree of approach of the approximate to the exact values. Therefore, we only apply it for the first half of the iterations, i.e., for $n = 1, \cdots, N/2$.

After determining approximate state values by the lookup table, we also apply the basis function approach for value function approximations; for instance, see [56] and [61] for further information on construction of basis functions. By considering a linear combination of objective functions introduced in (3), the value function $\overline{V}_t^b(\mathbf{S}^t)$ at state $\mathbf{S}^t$ can be written as

$$\overline{V}_t^b(\mathbf{S}^t) = \psi_1.\Big(\alpha\sum_{i=1}^{n^t} w_i^t p_i\Big) + \psi_2.\Big((1-\alpha)\sum_{i=1}^{n^t} w_i^t(T - a_i)\Big) \tag{6}$$

where parameters ψ_1 and ψ_2 are determined by the linear regression. In our computational experiments, we obtain the value function approximation with this basis function for states that are not visited by the algorithm (i.e., not existing in the lookup table). We observe that the best fitting basis function approximation is achieved when $\psi_1 = \psi_2 = \psi$.

5. Computational Experiments

In this section, we demonstrate performance of the stochastic dynamic programming model developed for the real-time surgery management problem using the proposed ADP algorithm. Some managerial insights are drawn from several computational experiments, including the performance comparison with a practical myopic heuristic and the analysis of different elective scheduling strategies. Algorithm 1 and the traditional dynamic programming algorithm (based on the backward value iteration technique) are implemented in Matlab. All numerical experiments are run on a PC with Windows 10 Enterprise operating system, CPU 4.00 GHz Intel Corei5 and 32 Gb of RAM. We now begin our discussion with the analysis of the computational performance of the proposed ADP algorithm.

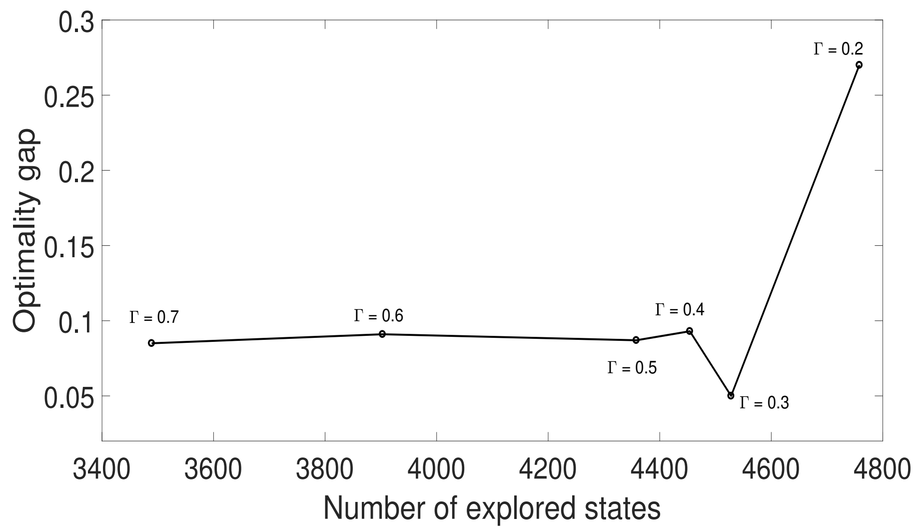
5.1. Computational Performance of the ADP Algorithm

The performance of the ADP algorithm is measured by two criteria: convergence speed and optimality gap. The *convergence speed (rate)* is defined as the number of iterations required to achieve the stability in state values. Following Hulshof et al., (2013), a state value is considered to be stable if it stays within 10% of the exact value. The *optimality gap* is defined as the average relative difference between the optimum and approximate values of each state visited by the algorithm. Note that, for the final states at the end of the decision horizon, the approximate and optimum values remain the same since they do not include evaluation of the future expected value functions. Thus, for a more rigorous comparison, we exclude the final states from the optimality gap calculation.

Since the optimality gap and, thus, optimum solution need to be obtained by the exact DP approach, in this section we consider a small problem instance which consists of $m = 2$ types of surgeries to be allocated into $R = 2$ operating rooms over $T = 6$ time periods. The random surgery durations of two surgery types follow simple uniform discrete distributions with the support of $\{1, 2\}$ and $\{2, 3\}$ for the first and second surgery types, respectively. There are $n_e = 5$ scheduled elective surgeries at the beginning of the decision horizon. The probability of one non-elective arrival at each time period is set at 25% with two surgery types being assigned randomly with equal probabilities. We assume that surgery priorities are set to 1 for all elective and non-elective patients in this instance. Finally, we chose $\alpha = 0.5$. The dynamic programming model for this small problem instance involves 104,224 possible states. The optimal policy is obtained by using the backward value iteration method within 3.2 hours. On the other hand, the ADP algorithm runs in less than a minute.

The performance of the ADP algorithm depends on parameters: β_n and Γ. The smoothing parameter β_n is defined as a linear function of iteration counter n as suggested by Powel (2009). From our empirical studies, we observe that the best convergence rate is achieved when β_n is set to a value varying between 0.1 and 0.3 at the initial step. They are then increased up to the maximum value of 0.9 in Algorithm 1. As mentioned earlier, in order to increase exploration within the algorithm, we incorporate randomness into the action selection procedure using parameter $\Gamma \in [0,1]$. According to Powel (2007), there is a trade-off between exploration and exploitation in the implementation of the ADP algorithm. The exploration is measured by the number of states explored while the exploitation is linked to the approximation of true state values which, in turn, affects the optimality gap. Figure 1 displays the number of states explored and the optimality gap at varying values of Γ. We observe that for this specific problem instance, the value of

Figure 1: Number of explored states and optimality gap obtained at different Γ levels



Γ is found as 0.3, which results in both a considerably large number of states explored and the smallest optimality gap. There is a 95% match between actions determined by the optimal and ADP based policies and the optimality gap is less than 5% with $\Gamma = 0.3$. We are going to use these settings of β_n and Γ in subsequent numerical experiments, which mainly aim to demonstrate the efficiency of the proposed ADP algorithm in larger real-life instances. Note that, in practice, for a particular problem instance, one should fine tune these parameters again, which will guarantee no worse results than those with pre-determined values of β_n and Γ.

We start with a base instance whose data are generated based on publicly available sources such as those described in the Surgery Scheduling Benchmark Set (2017). A detailed description of data is presented in Table 3. Similar to Jung et al. (2017), we consider the daily total operating time of 10 hours with $T = 20$ half-hour periods as the planning horizon. The number of rooms is set to be $R = 3$. The elective list is created such that all surgery types, three of them ($m = 3$), are equally represented with 3 surgeries per type for this instance. These settings are similar to those described in Jung et al. (2017) for trauma and neurology cases. We assume that random surgery durations follow log-normal distribution (see, e.g., May et al., 2000; Strum et al., 1998; and Strum et al., 2000).

Table 3: Detailed description of data

Description of Parameters	**Set Values**	**References**
Planning horizon: T time periods	20 epochs	Jung et al. (2017)
Length of time period: ΔT	30 minutes	Jung et al. (2017)
Surgery types: m	3	Jung et al. (2017)
Random surgery durations	Log-normal	May et al. (2000), Strum et al. (1998), Strum et al. (2000)
Type 1: mean (standard deviation)	1.5 (1.7)	
Type 2: mean (standard deviation)	2.5 (1.9)	Surgery Scheduling Benchmark Set (2017)
Type 3: mean (standard deviation)	3.8 (1.9)	
Electives priority (probability)	1 (100%)	
Non-electives priority (probability)	2 (80%)	Heng and Wright (2013), Van Riet et al. (2015)
Non-electives priority (probability)	3 (20%)	
Non-elective arrivals		
1 per time period: probability	12.5%	Paul and MacDonald (2012)
2 per time period: probability	4%	

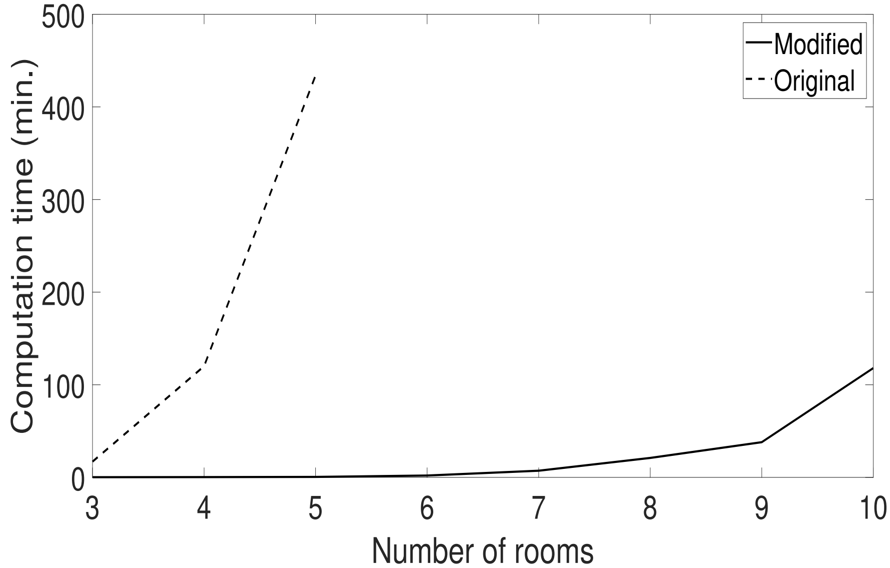
The first two moments of the log-normal distribution of each surgery type are estimated from data obtained from the Surgery Scheduling Benchmark Set (2017). There are three levels of priority for surgeries (denoted as 1, 2 and 3) as discussed in Paul and MacDonald (2012). All electives are set with the lowest level of priority at 1. The priority of non-electives is assumed to be either 2 or 3 due to their urgency. Following the discussion of priority in Heng and Wright (2013) and Van Riet et al. (2015), we fix the probability of a non-elective arrival with a certain priority to 80% and 20% for levels 2 and 3, respectively. As proposed by Paul and MacDonald (2012), the probability of having one non-elective arrival in a half-hour period is set to be 12.5% and that of two non-elective arrivals is 4%. Various literature sources indicate that distributions of non-elective and elective surgery types are very close to each other (see, for example, Wullink et al. (2007), Lamiri et al. (2007), and Ferrand et al. (2014a)). Therefore, we assume that types of non-elective arrivals are also equally distributed, i.e., 1/3 probability for each surgery type. Finally, we assume that all rooms can handle surgeries of all types in this base instance.

We run the proposed ADP algorithm and it takes 17 minutes to solve the base instance. We vary the number of operating rooms to check the performance of the algorithm. The results show that it takes more than 400 minutes for five rooms and we could not solve the problem instances with ten rooms. This is due to the exponential increase of the action space in each time period when the number of operating rooms is increased. In order to improve the computational efficiency of the proposed algorithm, we make the following observation. The decision to determine which patient to assign to a particular operating room only depends on their arrival times, priorities, and surgery types. This means that if there is more than one patient with the same arrival time, priority, and surgery type, then any of them can be selected as one of the (multiple) solutions with the same objective value. The state space of the proposed algorithm currently depends on the number of patients. Given the observation, if the number of different combinations of priorities and surgery types is smaller than the number of arrival patients within a time period, one can modify the state space so that it depends on the number of different combinations of priorities and surgery types instead of the number of patients. A brief summary of the procedure is as follows.

The surgeries can be clustered based on three main features: the arrival times of patients, priorities, and types of surgeries. The elective surgeries are assumed to be ready at the beginning of the day of surgery. In other words, their arrival times, as well as their priorities, are the same. As they only differentiate according to types of surgeries, we can group elective patients into three clusters based on their features. However, each non-elective surgery is considered as a single cluster by itself since each individual patient possesses surgeries with different types and priorities. With this change, the first three 'surgery' indices can be devoted to the three surgery clusters. In this case, the indicator variable ω_i^t (representing whether a surgery i is still waiting at time t, or not) can, instead, be interpreted as the 'number of elective surgeries waiting in that cluster'. The state variable is updated simply by decreasing its value by one (instead of setting it to 0 as in Section 3) if the decision is to assign a surgery from that cluster to the available room. Thus, we have $\omega_{x_r^t}^{t+1} = \omega_{x_r^t}^{t} - 1$. The remaining parts of the model including the action space remain the same. With this minor change in implementation of the algorithm, we can illustrate the performance of the ADP algorithm by solving even larger problem instances with different numbers of operating rooms.

Figure 2 shows the computational time (minutes) taken to solve the underlying dynamic program with different numbers of operating rooms using the two ways of modelling. As can be seen from Figure 2, the ADP algorithm takes two hours to solve the dynamic surgery scheduling problem with 10 operating rooms. We should also report that it takes 27 seconds to solve the base problem while the problem instance with 5 operating rooms can be solved in less than one minute. These results show that the proposed ADP algorithm can handle real-life instances efficiently.

Figure 2: Computation times taken to solve the surgery management problem with different number of operating rooms



5.2. Performance of ADP-based Policies

In practice, myopic policies are often used due to their simplicity. The myopic strategy is to select a greedy action based on a simple measure at any time. In surgery management, one needs to decide how to assign surgeries to available operating rooms. Mullen (2003) provides an exhaustive list of different assignment strategies applied for elective and non-elective patients. One considered measure is a priority measure computed as a function of waiting time and clinical scores. Similarly, Levtzion et al. (2010) consider another priority measure using both medical urgency and waiting time, which is applied in some hospitals. In order to demonstrate the effectiveness of the ADP-based policy, we first compare its performance with that of a myopic policy designed for the proposed model. If a surgery i is not assigned to the currently available room, the contribution to the total cost would be its priority-weighted waiting time and, potentially, the cost of a cancelled surgery. Therefore, we use the weighted cost measure as $c_m^t(i) = \alpha \cdot p_i + (1-\alpha)p_i(t-a_i)$ for the myopic policy and at any time t, the surgery with the highest cost will be assigned to the currently available operating room.

In order to compare two policies, we generate 1000 different scenarios with random non-elective arrivals and random surgery durations. Two performance measures are the total cumulative priority-weighted *waiting time* and the priority-weighted sum of *completed surgeries*, which can be derived from the two objective components, $\sum_{t=1}^{T} c_w^t(\mathbf{S}^t)$ and $c_d(\mathbf{S}^T)$, respectively. Table 4 displays the mean and standard error of these two performance measures obtained by the ADP-based and myopic policies.

As shown in these results, the ADP-based policy is statistically significantly better than the myopic

Table 4: Performance measures of the ADP-based and myopic policies

Performance Metric	ADP-based Policy	Myopic Policy
Waiting times	89.19 ± 0.79	96.7 ± 0.54
Completed surgeries	8.81 ± 0.07	7.67 ± 0.09

policy in terms of both waiting time and completed surgeries. This demonstrates the effectiveness of the ADP-based policy, which takes into account the effect of random arrivals in future periods and random surgery durations, as compared to the myopic policies usually adopted in practice. For example, unlike the myopic policy, the ADP-based policy might choose to assign a surgery of low priority instead of another surgery of higher priority if there is a chance that the other surgery cannot be scheduled and completed by the end of the day.

If the ADP-based policy is adopted instead of myopic policies, it is important for a decision-maker to be able to determine model parameters appropriately, especially parameter α which represents the trade-off between two performance measures, waiting time and completed surgeries. In order to show the effect of α on performance metrics, we run the ADP algorithm with different values of α.

Figure 3: Performance measures obtained by the ADP-based policy with different α levels

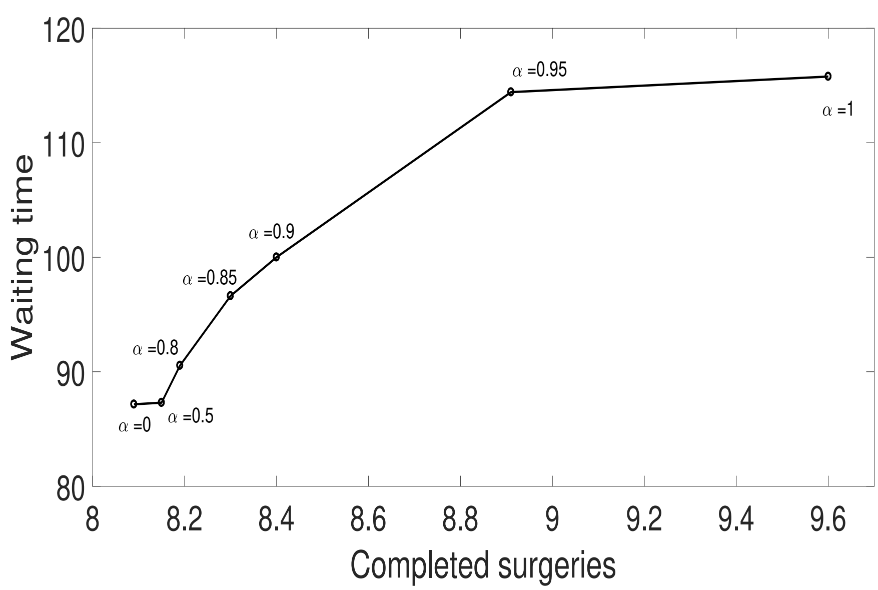
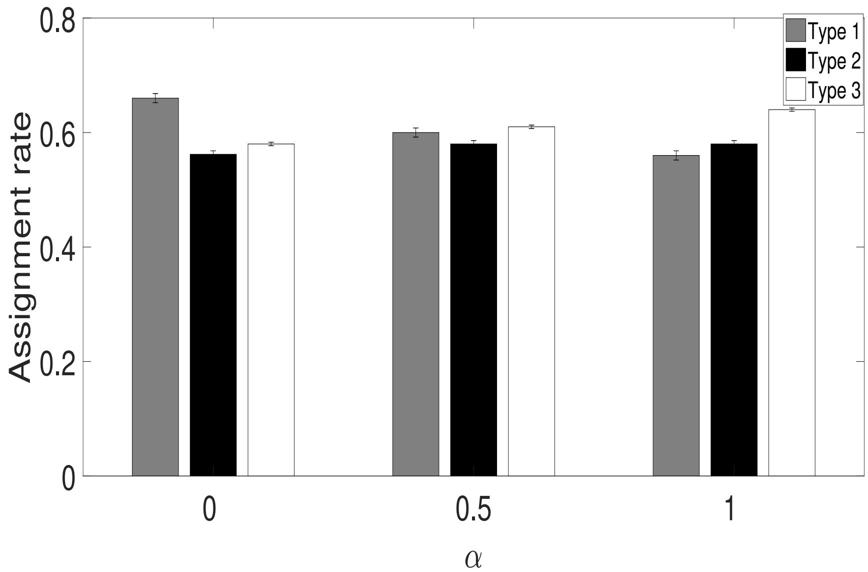


Figure 3 presents the two performance measures for different values of α. These results can support the decision-makers to set α appropriately based on the perceived importance of the two performance measures. Applying the common L-curve method used to select regularization parameters, one might choose $\alpha = 0.5$ since it is the “corner point” with the maximal curvature (see, for example, Hansen (2010)).

Parameter α affects not only performance measures but also the resulting policies. Figure 4 illustrates how the surgery assignment of different surgery types changes when α varies. We observe that when α increases from 0 to 1, the assignment rate, i.e., the percentage of assigned patients, of Type 1 surgeries with the shortest mean duration decreases while that of the other two types with longer mean durations increases. This shows that when the performance measure of completed surgeries is more important than

Figure 4: Assignment rates for different surgery types obtained by the ADP-based policy with varying α levels



waiting times, the resulting policies seem to try to assign less surgeries with shorter durations and more surgeries with longer durations. This is somewhat counter-intuitive, which indicates that the proposed ADP approach is able to take into account multiple factors including the randomness in surgery durations and surgery priority together to produce more effective policies.

The proposed ADP approach for the surgery management problem is not only effective as demonstrated, but also flexible. There are many different settings in practice that can be incorporated into the model. We focus on the three common settings, as block scheduling, separate elective/non-elective scheduling, and elective pre-scheduling to show their effects on performance measures.

Block Scheduling: In practice hospitals employ various scheduling strategies to improve their operational efficiency and patient satisfaction. Block scheduling is a type of scheduling with which an operating room can be blocked for some particular surgery types for a certain period (Conforti et al., 2010). For instance, it is quite common for surgeries with long durations and those requiring special settings; for example, there might be an operating room reserved only for trauma and orthopaedic surgeries.

For this experiment, we modify the base instance so that operating rooms 2 and 3 are blocked only for type 2 and type 3 surgeries, respectively. On the other hand, operating room 1 can accommodate both types 1 and 3 surgeries. This results in a modified model with smaller feasible sets $\mathcal{X}_r^t$ for scheduling of each operating room r at time t. Table 5 shows the performance measures of the two settings, block scheduling and no block (base instance) scheduling. Note that with block scheduling constraints, performance measures are not as good as those of the base instance in which every room can accommodate all types of surgeries. In other words, the base scheduling (with no block setting) provides more completed surgeries and less waiting times. The result indicates that one should aim to increase the flexibility of operating rooms as much as possible to improve the overall performance. With respect to computational time, given the reduction in sizes of feasible sets, the block scheduling instance can be solved much faster; the computational time drops by more than half in this case.

Table 5: Performance of different scheduling strategies depending on surgery types allocated to rooms

Room Setting for Surgery	**Waiting Time**	**Completed Surgeries**
Block scheduling	97.97 ± 0.5	7.47 ± 0.092
Base (no block) scheduling	89.19 ± 0.79	8.81 ± 0.07

Separate Elective/Non-elective Scheduling: In practice, some hospitals might separate elective and non-elective surgeries and devote different operating rooms for their surgeries (e.g., see Van Riet and Demeulemeester, 2015). In order to test this strategy, we design a new experiment by modifying the base instance. We assume that room 3 is reserved for non-elective surgeries only. We also reduce the number of electives to 6 in both instances given that electives in the modified instance can only be operated in two rooms, not three. This again results in a modified model with smaller feasible set $\mathcal{X}_r^t$. Table 6 shows the performance measures of the two settings with separate elective/non-elective scheduling and no separation (base instance). Once again, the restriction on the allocation of surgeries to specific operating rooms does not make the performance measures better even though it helps to reduce the computational time due to smaller feasible sets. This implies that one should consider the option of allowing non-elective surgeries to be performed in as many operating rooms as possible.

Table 6: Performance measures obtained by different policies using separate elective/non-elective surgery scheduling instance vs base instance (under no separation)

Room Setting for Surgery	**Waiting Time**	**Completed Surgeries**
Separate elective/non-elective surgery	76.31 ± 1.23	5.23 ± 0.04
Base instance (no separation)	40.13 ± 1.06	6.47 ± 0.04

Elective Pre-scheduling: Elective schedules can be pre-determined in practice. Hospitals apply various strategies to assign elective surgeries to different operating rooms; for example, the *longest case first* rule (Denton et al., 2007). In this case, when an operating room is available, surgeries are chosen from the current first-in-line elective surgeries (one for each operating room) and any non-electives which are currently waiting. We call this strategy *fixed elective scheduling*. Another strategy using elective pre-scheduling which is adopted in some hospitals is called *fixed room scheduling* or *no swapping scheduling* (Pulido et al., 2014). Under this strategy, when an operating room becomes available, surgeries are chosen among electives pre-assigned to that room and any non-electives currently waiting. In this experiment, we implement both strategies and compare their performances with *flexible scheduling (base instance)* in terms of waiting time and completed surgeries. Note that both of these settings again reduce the size of feasible sets $\mathcal{X}_r^t$. Table 7 presents the performance measures of the three settings.

Table 7: Comparison of performance measures of elective pre-scheduling instances vs flexible scheduling

Elective Scheduling Setting	Waiting Time	Completed surgeries
Fixed elective scheduling	129.67 ± 1.58	5.91 ± 0.02
Fixed room scheduling	94.07 ± 0.55	8.72 ± 0.06
Flexible scheduling (base case)	89.19 ± 0.79	8.81 ± 0.07

These results indicate that flexible scheduling performs better than any fixed scheduling strategy. Together with other results, this demonstrates the flexibility of the proposed approach in generating effective policies for surgery management under different practical conditions.

6. Summary and Concluding Remarks

In this paper, we introduce a stochastic dynamic programming model for the real-time surgery management problem under random non-elective arrivals and stochastic surgery durations. The underlying decision-making process requires to take real-time decisions related to assigning a surgery to an operating room as soon as it becomes available during the day. We propose a simulation-based ADP algorithm to tackle the curse of dimensionality of the stochastic dynamic programming model. In particular, we apply the lookup table and double-pass approaches to obtain approximately optimum policy for real-sized instances. The numerical results of a small problem instance show that the exact and approximate policies coincide in 95% of the states of the system.

We also design computational experiments using a real data set and consider the myopic heuristic as a benchmark for the performance comparison of the proposed ADP approach. We derive some managerial insights to illustrate how the proposed approach can help to improve the operational efficiency of surgery management. The computational numerical results demonstrate that the proposed ADP approach for the surgery management problem is effective and also flexible.

For future work, the modelling framework introduced in this paper can be extended to a longer planning horizon than a day. In this case, the problem complexity further increases since the decisions made on admission of a patient today will have an impact on the next day's schedules. This requires effective and efficient solution methods based on approximation approaches to reduce the state and action spaces of the system.

References

[1] Addis, B., Carello, G., Grosso, A., Tànfani, E., 2016. Operating room scheduling and rescheduling: a rolling horizon approach. Flexible Services and Manufacturing Journal 28, 206–232. URL: http://dx.doi.org/10.1007/s10696-015-9213-7, doi:10.1007/s10696-015-9213-7.

[2] Allmendinger, R., Ehrgott, M., Gandibleux, X., Geiger, M.J., Klamroth, K., Luque, M., 2017. Navigation in multiobjective optimization methods. Journal of Multi-Criteria Decision Analysis 24, 57–70.

[3] Aloulou, M.A., Portmann, M.C., 2003. An efficient proactive reactive scheduling approach to hedge against shop floor disturbances, in: In Proceedings of the 1 st Multidisciplinary International Conference on Scheduling: Theory and Applications, MISTA 2003, Citeseer. p. 1.

[4] Astaraky, D., Patrick, J., 2015. A simulation based approximate dynamic programming approach to multi-class, multi-resource surgical scheduling. European Journal of Operational Research 245, 309–319. URL: http://www.sciencedirect.com/science/article/pii/S0377221715001332, doi:10.1016/j.ejor.2015.02.032.

[5] Bandi, C., Gupta, D., 2020. Operating Room Staffing and Scheduling. Manufacturing & Service Operations Management .

[6] Batun, S., Denton, B.T., Huschka, T.R., Schaefer, A.J., 2011. Operating room pooling and parallel surgery processing under uncertainty. INFORMS journal on Computing 23, 220–237.

[7] Billaut, J.C., Roubellat, F., 1996. A new method for workshop real time scheduling. International Journal of Production Research 34, 1555–1579.

[8] Blake, J.T., Dexter, F., Donald, J., 2002. Operating room managers' use of integer programming for assigning block time to surgical groups: a case study. Anesthesia & Analgesia 94, 143–148.

[9] Borgman, N.J., 2017. Managing urgent care in hospitals. Ph.D. thesis. University of Twente.

[10] Boyan, J.A., Littman, M.L., 2000. Exact solutions to time-dependent MDPs, in: NIPS, pp. 1026–1032.

[11] Campbell, D., Arnett, J., 2015. A&E meltdown forces thousands of operations to be cancelled. The Guardian URL: http://www.theguardian.com/society/2015/jan/10/a-and-e-doctors-warn-patient-misery-planned-surgery.

[12] Cardoen, B., Demeulemeester, E., Beliën, J., 2010. Operating room planning and scheduling: A literature review. European Journal of Operational Research 201, 921–932.

[13] Carey, K., Burgess, J.F., Young, G.J., 2011. Hospital competition and financial performance: the effects of ambulatory surgery centers. Health Economics 20, 571–581. URL: http://dx.doi.org/10.1002/hec.1617, doi:10.1002/hec.1617.

[14] Conforti, D., Guerriero, F., Guido, R., 2010. A multi-objective block scheduling model for the management of surgical operating rooms: new solution approaches via genetic algorithms, in: Health Care Management (WHCM), 2010 IEEE Workshop on, IEEE. pp. 1–5.

[15] Denton, B., Viapiano, J., Vogl, A., 2007. Optimization of surgery sequencing and scheduling decisions under uncertainty. Health care management science 10, 13–24.

[16] Duma, D., Aringhieri, R., 2015. An online optimization approach for the Real Time Management of operating rooms. Operations Research for Health Care 7, 40–51.

[17] Duma, D., Aringhieri, R., 2018. The real time management of operating rooms, in: Operations research applications in health care management. Springer, pp. 55–79.

[18] Erdem, E., Qu, X., Shi, J., 2012. Rescheduling of elective patients upon the arrival of emergency patients. Decision Support Systems 54, 551–563.

[19] Erdogan, S.A., Denton, B.T., Cochran, J.J., Cox, L.A., Keskinocak, P., Kharoufeh, J.P., Smith, J.C., 2011. Surgery planning and scheduling.

[20] van Essen, J.T., Hurink, J.L., Hartholt, W., van den Akker, B.J., 2012. Operating room rescheduling. URL: http://eprints.eemcs.utwente.nl/21177/01/wp{_}368.pdf.

[21] Ferrand, Y.B., Magazine, M.J., Rao, U.S., 2014. Partially flexible operating rooms for elective and emergency surgeries. Decision Sciences 45, 819–847.

[22] Freeman, N.K., Melouk, S.H., Mittenthal, J., 2016. A scenario-based approach for operating theater scheduling under uncertainty. Manufacturing & Service Operations Management 18, 245–261.

[23] Fregene, T., Wintle, S., Raman, V.V., Edmond, H., Rizvi, S., 2017. Making the experience of elective surgery better. BMJ Open Qual 6, e000079.

[24] Gerchak, Y., Gupta, D., Henig, M., 1996. Reservation planning for elective surgery under uncertain demand for emergency surgery. Management Science 42, 321–334.

[25] Guerriero, F., Guido, R., 2011. Operational research in the management of the operating theatre: a survey. Health Care Management Science 14, 89–114.

[26] Guido, R., Conforti, D., 2017. A hybrid genetic approach for solving an integrated multi-objective operating room planning and scheduling problem. Computers & Operations Research 87, 270–282.

[27] Gupta, D., Wang, L., 2008. Revenue management for a primary-care clinic in the presence of patient choice. Operations Research 56, 576–592.

[28] Gür, S., Eren, T., 2018. Application of operational research techniques in operating room scheduling problems: literature overview. Journal of Healthcare Engineering 2018.

[29] Hansen, K.B., Vilsboll, T., Bagger, J.I., Holst, J.J., Knop, F.K., 2010. Reduced glucose tolerance and insulin resistance induced by steroid treatment, relative physical inactivity, and high-calorie diet impairs the incretin effect in healthy subjects. The Journal of clinical endocrinology and metabolism 95, 3309–3317. doi:10.1210/jc.2010-0119.

[30] of Health, D., 2016. Department of Health:Referral to Treatment Statistics. URL: https://www.gov.uk/government/publications?keywords=referral+to+treatment{&}publication{_}filter{_}option=all{&}topics{%}5B{%}5D=all{&}departments{%}5B{%}5D=all{&}official{_}document{_}status=all{&}world{_}locations{%}5B{%}5D=all{&}from{_}date={&}to{_}date=.

[31] Healthcare Financial Management Association (HFMA), 2005. Achieving operating room efficiency through process integration. Technical Report. Healthcare Financial Management Association.

[32] Heng, M., Wright, J.G., 2013. Dedicated operating room for emergency surgery improves access and efficiency. Canadian Journal of Surgery 56, 167.

[33] Heydari, M., Soudi, A., 2016. Predictive/Reactive Planning and Scheduling of a Surgical Suite with Emergency Patient Arrival. Journal of medical systems 40, 1–9.

[34] Hooshmand, F., MirHassani, S.A., Akhavein, A., 2018. Adapting GA to solve a novel model for operating room scheduling problem with endogenous uncertainty. Operations Research for Health Care 19, 26–43.

[35] Hosseini, N., 2012. Managing Elective and Non-elective Case Assignments for an Operating Room Suite. Ph.D. thesis. Clemson University. URL: http://tigerprints.clemson.edu/cgi/viewcontent.cgi?article=2065{&}context=all{_}dissertations{&}sei-redir=1{&}referer=http{%}3A{%}2F{%}2Fscholar.google.com.tr{%}2Fscholar{%}3Fstart{%}3D60{%}26q{%}3Drelated{%}3A1S81VZWd64WfYM{%}3Ascholar.google.com{%}2F{%}26hl{%}3Dtr{%}26as{_}sdt{%}3D0{%}2C5{#}search={%}22related{%}3A1S81VZWd64WfYM{%}3Ascholar.google.com{%}2F{%}22.

[36] Hulshof, P.J.H., Mes, M.R.K., Boucherie, R.J., Hans, E.W., 2016. Patient admission planning using approximate dynamic programming. Flexible Services and Manufacturing Journal 28, 30–61.

[37] Jung, K.S., Pinedo, M., Sriskandarajah, C., Tiwari, V., 2017. Scheduling Operating Rooms with Elective and Emergency Surgeries. Available at SSRN 2965730 .

[38] Jung, K.S., Pinedo, M., Sriskandarajah, C., Tiwari, V., 2019. Scheduling Elective Surgeries with Emergency Patients at Shared Operating Rooms. Production and Operations Management 28, 1407–1430.

[39] Kamran, M.A., Karimi, B., Dellaert, N., 2020. A column-generation-heuristic-based benders' decomposition for solving adaptive allocation scheduling of patients in operating rooms. Computers & Industrial Engineering 148, 106698.

[40] Khalfalli, M., Abdelaziz, F.B., Kamoun, H., 2019. Multi-objective surgery scheduling integrating surgeon constraints. Management Decision .

[41] Lamiri, M., Dréo, J., Xie, X., 2007. Operating room planning with random surgery times, in: Automation Science and Engineering, 2007. CASE 2007. IEEE International Conference on, IEEE. pp. 521–526.

[42] Lamiri, M., Xie, X., Dolgui, A., Grimaud, F., 2008. A stochastic model for operating room planning with elective and emergency demand for surgery. European Journal of Operational Research 185, 1026–1037.

[43] Levtzion-Korach, O., Murphy, K.G., Madden, S., Dempsey, C., 2010. For urgent and emergent cases, which one goes to the OR first? OR manager 26, 1,11–13.

[44] Macario, A., Vitez, T.S., Dunn, B., McDonald, T., 1995. Where are the costs in perioperative care?: Analysis of hospital costs and charges for inpatient surgical care. Anesthesiology 83, 1138–1144.

[45] Maghzi, P., Roohnavazfar, M., Mohammadi, M., Naderi, B., 2019. A Mathematical Model for Operating Room Scheduling Considering Limitations on Human Resources Access and Patient Prioritization. Journal of Quality Engineering and Production Optimization 4, 67–82.

[46] May, J.H., Strum, D.P., Vargas, L.G., 2000. Fitting the lognormal distribution to surgical procedure times. Decision Sciences 31, 129–148.

[47] Megow, N., Uetz, M., Vredeveld, T., 2006. Models and algorithms for stochastic online scheduling. Mathematics of Operations Research 31, 513–525.

[48] Miao, H., Wang, J.J., 2021. Scheduling elective and emergency surgeries at shared operating rooms with emergency uncertainty and waiting time limit. Computers & Industrial Engineering 160, 107551.

[49] Min, D., Yih, Y., 2014. Managing a patient waiting list with time-dependent priority and adverse events. RAIRO-Operations Research 48, 53–74.

[50] Möhring, R.H., 2000. Scheduling under uncertainty: Optimizing against a randomizing adversary, in: International Workshop on Approximation Algorithms for Combinatorial Optimization, Springer. pp. 15–26.

[51] Moreno, M.S., Blanco, A.M., 2018. A fuzzy programming approach for the multi-objective patient appointment scheduling problem under uncertainty in a large hospital. Computers & Industrial Engineering 123, 33–41.

[52] Mullen, P.M., 2003. Prioritising waiting lists: how and why? European Journal of Operational Research 150, 32–45. URL: `http://www.sciencedirect.com/science/article/pii/S0377221702007798`, doi:`https://doi.org/10.1016/S0377-2217(02)00779-8`.

[53] Ozen, A., Marmor, Y., Rohleder, T., Balasubramanian, H., Huddleston, J., Huddleston, P., 2016. Optimization and simulation of orthopedic spine surgery cases at Mayo Clinic. Manufacturing & Service Operations Management 18, 157–175.

[54] Ozkarahan, I., 2000. Allocation of surgeries to operating rooms by goal programing. Journal of Medical Systems 24, 339–378.

[55] Paul, J.A., MacDonald, L., 2012. Determination of number of dedicated OR's and supporting pricing mechanisms for emergent surgeries. Journal of the Operational Research Society 64, 912–924.

[56] Powell, W.B., 2007. Approximate Dynamic Programming: Solving the Curses of Dimensionality. volume 703. John Wiley & Sons.

[57] Powell, W.B., 2009. What you should know about approximate dynamic programming. Naval Research Logistics 56, 239–249. doi:10.1002/nav.20347.

[58] Pulido, R., Aguirre, A.M., Ortega-Mier, M., García-Sánchez, Á., Méndez, C.A., 2014. Managing daily surgery schedules in a teaching hospital: a mixed-integer optimization approach. BMC Health Services Research 14, 464.

[59] Rahimi, I., Gandomi, A.H., 2021. A comprehensive review and analysis of operating room and surgery scheduling. Archives of Computational Methods in Engineering 28, 1667–1688.

[60] Schofield, W.N., Rubin, G.L., Piza, M., Lai, Y.Y., Sindhusake, D., Fearnside, M.R., Klineberg, P.L., 2005. Cancellation of operations on the day of intended surgery at a major Australian referral hospital. Med J Aust 182, 612–615.

[61] Sierra-García, J.E., Santos, M., 2021. Lookup Table and Neural Network Hybrid Strategy for Wind Turbine Pitch Control. Sustainability 13, 3235.

[62] Silva, T.A., de Souza, M.C., 2020. Surgical scheduling under uncertainty by approximate dynamic programming. Omega (United Kingdom) 95, 102066. doi:10.1016/j.omega.2019.05.002.

[63] Spangenberg, N., Wilke, M., Augenstein, C., Franczyk, B., 2018. Online Surgery Rescheduling-A Data-driven Approach for Real-time Decision Support., in: ICEIS (1), pp. 336–343.

[64] Spratt, B., Kozan, E., 2019. A real-time reactive framework for the surgical case sequencing problem. Flexible Services and Manufacturing Journal , 1–29.

[65] Spratt, B., Kozan, E., 2021. A real-time reactive framework for the surgical case sequencing problem. Flexible Services and Manufacturing Journal 33, 183–211.

[66] Strum, D.P., May, J.H., Vargas, L.G., 1998. Surgical procedure times are well modeled by the lognormal distribution. Anesthesia & Analgesia 86, 47S–47S.

[67] Strum, D.P., May, J.H., Vargas, L.G., 2000. Modeling the Uncertainty of Surgical Procedure TimesComparison of Log-normal and Normal Models. The Journal of the American Society of Anesthesiologists 92, 1160–1167.

[68] Stuart, K., Kozan, E., 2012. Reactive scheduling model for the operating theatre. Flexible Services and Manufacturing Journal 24, 400–421.

[69] Sun, Y., Li, X., 2013. Response surface optimisation of surgery start times in a single operating room using designed simulation experiments. International Journal of Healthcare Technology and Management 14, 61–72.

[70] University of Twente, 2017. Surgery Scheduling Benchmark Set. URL: `https://www.utwente.nl/en/choir/research/benchmarkORscheduling/`.

[71] Van Riet, C., Demeulemeester, E., 2015. Trade-offs in operating room planning for electives and emergencies: A review. Operations Research for Health Care 7, 52–69.

[72] Vandenberghe, M., Vuyst, S.D., Aghezzaf, E.H., Bruneel, H., 2021. Stochastic surgery selection and sequencing under dynamic emergency break-ins. Journal of the Operational Research Society 72, 1309–1329.

[73] Wang, B., Han, X., Zhang, X., Zhang, S., 2015. Predictive-reactive scheduling for single surgical suite subject to random emergency surgery. Journal of Combinatorial Optimization 30, 949–966.

[74] Wu, S.D., Byeon, E.S., Storer, R.H., 1999. A graph-theoretic decomposition of the job shop scheduling problem to achieve scheduling robustness. Operations Research 47, 113–124.

[75] Wullink, G., Van Houdenhoven, M., Hans, E.W., van Oostrum, J.M., van der Lans, M., Kazemier, G., 2007. Closing emergency operating rooms improves efficiency. Journal of Medical Systems 31, 543–546.

[76] Xiao, G., van Jaarsveld, W., Dong, M., van de Klundert, J., 2018. Models, algorithms and performance analysis for adaptive operating room scheduling. International Journal of Production Research 56, 1389–1413.

[77] Zhang, J., Dridi, M., El Moudni, A., 2019a. A Markov decision model with dead ends for operating room planning considering dynamic patient priority. RAIRO-Operations Research 53, 1819–1841.

[78] Zhang, J., Dridi, M., El Moudni, A., 2019b. A two-level optimization model for elective surgery scheduling with downstream capacity constraints. European Journal of Operational Research 276, 602–613.

[79] Zhang, Z., Xie, X., Geng, N., 2014. Dynamic Surgery Assignment of Multiple Operating Rooms With Planned Surgeon Arrival Times. IEEE Transactions on Automation Science and Engineering 11, 680–691.