European Journal of Operational Research 270 (2018) 734–746

Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Innovative Applications of O.R.

A sequential stochastic mixed integer programming model for tactical master surgery scheduling

Ashwani Kumar*, Alysson M. Costa, Mark Fackrell, Peter G. Taylor

School of Mathematics and Statistics, The University of Melbourne, Parkville, Vic 3010, Australia

ARTICLE INFO

Article history:
Received 16 October 2017
Accepted 7 April 2018
Available online 13 April 2018

Keywords:
OR in health services
Patient flow
Stochastic scheduling
Elective surgery
Tactical master surgery schedule

ABSTRACT

In this paper, we develop a stochastic mixed integer programming model to optimise the tactical master surgery schedule (MSS) in order to achieve a better patient flow under downstream capacity constraints. We optimise the process over several scheduling periods and we use various sequences of randomly generated patients' length of stay scenario realisations to model the uncertainty in the process. This model has the particularity that the scenarios are chronologically sequential, not parallel. We use a very simple approach to enhance the non-anticipative feature of the model, and we empirically demonstrate that our approach is useful in achieving the desired objective. We use simulation to show that the most frequently optimal schedule is the best schedule for implementation. Furthermore, we analyse the effect of varying the penalty factor, an input parameter that decides the trade-off between the number of cancellations and occupancy level, on the patient flow process. Finally, we develop a robust MSS to maximise the utilisation level while keeping the number of cancellations within acceptable limits.

© 2018 Elsevier B.V. All rights reserved.

1. Introduction

A major challenge for the healthcare industry in Australia and across the world is to keep up with the growing demand for services. In Australia, public hospitals maintain elective surgery waiting lists, and hospital management aims to maximise the number of operations performed.

In order to achieve this goal, we need to analyse and optimise patient flow in the surgical suite in such a way that we can get maximum throughput. Some researchers have modelled patient flow as a queuing network, (see Cote, 2000). However, there are many disadvantages in modelling elective patient flow as a queuing network. First, the elective patient arrival process is under management's control, and it can be adjusted to obtain better patient flow. Second, there are some operational restrictions on the arrival process, for example no elective patients are operated on during weekends or holidays, whereas downstream resources are still available. Although we can develop a discrete event simulation model to analyse the elective patient flow process, it is difficult to schedule elective patients' operations optimally in order to obtain a better patient flow. In simulation models, we select an arrival process and a service process, and we observe the behaviour of the system. We may improve the system by adjusting the arrival process intuitively or with simulation optimisation techniques (see Figueira & Almada-lobo, 2014), but it can be difficult to obtain the optimal arrival process.

To develop a deterministic optimal schedule, we can model the patient flow process as a flow shop scheduling problem where each patient represents a job, each set of resources represents a machine, and a patient's length of stay (LoS) represents its service time, (see Pham & Klinkert, 2008). However, as we discuss below, there are some important aspects of the patient flow process that are difficult to capture in a flow shop model. As a result, the schedule obtained cannot ensure smooth patient flow. We propose a mixed integer programming (MIP) model because it can incorporate scenarios to mimic the uncertainty in the process, and it has flexibility that allows us to model other essential characteristics of the patient flow. The main characteristics of the patient flow process are

- The LoS is random and we only know its distribution. The range of LoS is quite large. For example, a patient may stay in an intensive care unit (ICU) after surgery from one day to 60 days. Moreover, LoS distributions are generally positively skewed and long-tailed.
- At no point in time are all the resources in a downstream facility available. For example, if we are developing a four-week schedule to optimise patient flow in the ICU, then on the first day of each four-week period there will be some patients from

* Corresponding author.
E-mail addresses: ashwanik@student.unimelb.edu.au (A. Kumar), alysson.costa@unimelb.edu.au (A.M. Costa), fackrell@unimelb.edu.au (M. Fackrell), taylorpg@unimelb.edu.au (P.G. Taylor).

https://doi.org/10.1016/j.ejor.2018.04.007
0377-2217/© 2018 Elsevier B.V. All rights reserved.

the previous periods occupying the ICU beds with an unknown remaining LoS. This means that not only is the LoS random, but the resource availability also has some uncertainty.
- In the actual process, there is a trade-off between resource utilisation and cancellations. For example, we can get a very high utilisation (say, $\geq 90\%$) by scheduling the same number of patients as the number of beds in a downstream facility each day. However, we will be required to cancel many elective operations each day because of capacity shortage. Therefore, in the real process, we need to maximise resource utilisation while keeping the cancellations within an acceptable limit.

We develop a MIP model to analyse and optimise the patient flow process that is capable of capturing all these unique characteristics. Uncertainty is incorporated by using various random LoS scenario realisations, and non-anticipation is imposed by constraining the model to schedule patients in the same order as their position in the queue. Moreover, a particularity of our model is that the scenario realisations are chronologically sequential and not parallel. The rest of the paper is structured as follows. In Section 2, we present a review of the literature on scheduling elective surgery. In Section 3, we describe the problem in detail. We discuss key concepts and our approach thoroughly in Section 4, and the MIP model is presented in Section 5. Next, we conduct some computational experiments and present our results in Section 6. We conclude the paper and present some directions for further research in Section 7.

2. Literature review

Strategic operating room management involves long term decision making such as designing a master surgical schedule (MSS). An MSS is usually a two or four-week timetable which repeats for several months in which surgeons from various surgical specialities are assigned to operating theatres. The first stage is a case mix planning in which available operating theatres' times are divided into various time blocks, and the required numbers of time blocks are allocated to various surgical departments. Researchers have developed linear programming and mixed integer programming models for case mix planning, (see Blake & Carter, 2002).

The second stage involves creating an MSS in which each time block is allocated to a surgical team. In the third stage, patients are booked for surgery. Usually, surgical liaison nurses talk to patients who are waiting for elective surgery and schedule their operations in suitable time slots. Surgical liaison nurses do this manually on a daily basis. Building an MSS is an important step in strategic planning, and it is a widely studied problem in the literature on the application of operations research in healthcare management. Various researchers have studied this problem from different aspects.

Rafaliya (2013) suggested a MIP model with the objective of reducing overutilisation and underutilisation of the recovery room. However, he assumed that the recovery time was deterministic. Min and Yih (2010) suggested a stochastic scheduling model with the objective of minimising the sum of overtime in operating theatres and patients' waiting time. Carter and Ketabi (2012) developed a MIP model with the objective of balancing daily bed demand in a ward. They fitted lognormal distributions to the LoS data in each surgical department and generated random LoS scenarios to use in their deterministic MIP model. They used a sampling average approximation (SAA) approach to obtain the final schedule. Adan, Bekkers, Dellaert, Vissers, and Yu (2009) developed a MIP model to minimise the deviation from a target resource utilisation level where the resources were available operating theatre hours, ICU bed hours, and ward bed hours. They computed the probability of a patient staying in the hospital on each day after surgery by using the empirical distribution of the LoS data. By using these probabilities, they computed the *expected* number of occupied beds on each day of the scheduling period and used this as a proxy for stochastic resource demand.

There is another strand of literature on developing an MSS with the objective of balancing bed demand in a downstream facility. For example, Chow, Puterman, Salehirad, Huang, and Atkins (2011) developed a MIP model to balance the *expected* number of ward beds used on a given day by minimising the maximum of the *expected* number of beds used over the scheduling period. Beliën and Demeulemeester (2007) developed a MIP model with the objective of minimising the shortage of *expected* ward bed utilisation and the variance of ward bed utilisation at a downstream facility. However, they made a simplifying assumption that the LoS followed a discrete distribution with a small range of possible values. Cappanera, Visintin, and Banditori (2014) investigated the effect of various scheduling strategies on the utilisation level and overbooking. They developed a MIP model in which they considered three different objectives to obtain different scheduling schemes. The objectives they considered were the minimisation of the maximum daily utilisation of ward beds, the minimisation of the difference between the maximum and the minimum daily utilisations of ward beds, and the minimisation of the sum of the quadratic overrun of operating theatres. They divided patients into various groups within a surgical department according to their LoS and their surgery duration, and they used the average value in each group as an input for their MIP model to obtain an optimal schedule.

The above-mentioned MIP models were either based on the assumption that the LoS is deterministic (Carter & Ketabi, 2012; Min & Yih, 2010) or were developed to balance the *expected* resource demand, or to minimise the *expected* shortage of resources (Adan et al., 2009; Beliën & Demeulemeester, 2007; Chow et al., 2011). As we will discuss in Section 3 and demonstrate in Section 6, these approaches may not provide us with the desired outcomes. We have developed a stochastic MIP model that balances the actual resource demand on each day and maximises the throughput while keeping the number of cancellations within limits. The novelty in our model is that it utilises the given LoS scenario realisations chronologically in a sequential manner, and not in a parallel manner. Moreover, we constrain our model to schedule patients in their queueing order and repeat the scheduling decision over a long time-horizon. This helps us incorporate the variability in the LoS without making any assumptions regarding the LoS distribution. The model is presented in Section 5. Next, we will describe the problem in detail.

3. Problem description

The problem we discuss here is driven from a real life situation faced by a hospital. Our partner in this research project is a major metropolitan public hospital located in Melbourne, Australia. The hospital management is interested in improving patient flow in the surgical suite (the surgical suite includes operating theatres, recovery rooms, the ICU, and surgical wards) so that they can deliver surgical services more efficiently. They have been frequently cancelling elective operations on the day of surgery because of a capacity shortage in the intensive care unit (ICU). Therefore, we were required to model and analyse the elective patient flow in the surgical suite, and develop a scheduling scheme to optimise the patient flow process.

In a surgical suite, patients receive service at one care step and move to a subsequent care step. The LoS at each care step is random and there is no provision for waiting. Patient flow is controlled by the bottleneck facility, that is, the facility with the minimum capacity in the surgical suite. However, because of the stochastic nature of the process, the bottleneck may change from

one care step to another according to the type of patients being operated on. For example, if we operate on many day surgery patients (the patients whose post operative LoSs are generally a few hours), then the operating theatres will act as a bottleneck facility because the demand for post-operative care services is reduced. Whereas if we operate on many complex patients with longer post operative recovery times, then the ICU or the ward will act as a bottleneck facility.

After interviewing hospital staff, we identified that the ICU is currently the bottleneck facility. An ICU is a multidisciplinary care facility where a wide range of critically ill patients are treated. It receives inpatient flow from operating theatres, the emergency department, critically ill patients in wards, and transfers from other hospitals. The elective surgery patients' arrivals are scheduled, whereas others, such as the emergency patients' arrivals, are random. Although elective patients' arrivals are scheduled, elective operations are cancelled frequently because of the unavailability of an ICU bed. On many occasions, the ICU reaches capacity due to the elective patient flow. In a flow network where one node acts as a bottleneck, we can optimise the overall flow by maximising the flow at that node.

From the literature review, it is clear that the stochastic nature of the LoS increases the complexity of the problem. In the healthcare industry, resources are bundled together and they are quantifiable. For example, a twenty bed ICU means the ICU is fully equipped with all the required machines, and staff to accommodate twenty patients. The LoS is an important measure because patient resource demand is approximated by their LoS. Some researchers have used average LoS and others have modelled the stochastic LoS by using its empirical distribution. A major problem with this approach is that an empirical distribution function is just an empirical measure of one sample. Our data indicated that there was around two percent of very complex patients who stay much longer than the others, and they can have a wide range of possible LoS values. Since the percentage of these patients is small, even a relatively large dataset cannot capture all the possible LoS values. To capture the randomness in the process appropriately, we need a distribution function which can generate various possible scenario realisations.

Moreover, balancing the *expected* resource demand in a downstream facility does not ensure optimal outcomes. When we balance the expected resource demand, we ignore the fact that the actual resource demand may vary drastically from the expected resource demand because patient resource demand has a large variation. The sample average resource demand (average resource demand across all ICU beds) may converge to the expected resource demand if the law of large numbers is applicable. This means that either we are averaging a very large number of patients' resource demands, or the variation in their resource demand is small, or both. However, in reality, the number of patients whose resource demand is being averaged is quite small, and the variation in patient resource demand is quite large. Because of this, the pooled resource demand is still a random variable with a reasonably large variance. Therefore, ensuring that the *expectation* of pooled resource demand on each day is less than or equal to the available resources is not sufficient.

Cappanera et al. (2014) used simulation to assess the quality of schedules generated by a MIP model with the objective of balancing the *expected* resource demand. They found that using the optimal schedule resulted in a large amount of overbooking. In a real life scenario, it is not possible to accommodate the overbooked patients by temporarily increasing resources. Instead, they are cancelled. However, the model optimises the arrival process based on the assumption that no patient is cancelled. As a result, the actual throughput decreases drastically when we take into account cancellations.

Furthermore, targeting a very low utilisation level may ensure very few cancellations. However, this will result in a poor throughput and an inefficient system. In a real life scenario, cancellations are unavoidable. It makes more sense to model the patient flow process with cancellations and optimise that process. Therefore, in our model, we propose explicit decision variables to model cancellations. In the next section, we will discuss key concepts and our approach to analyse and optimise the patient flow process.

4. Fundamental concepts and our approach

To make our approach realistic and useful, we need to model and optimise the process with a similar level of uncertainty as that of the actual process. We develop a strategy to keep a comparable level of uncertainty in the MIP model. Kumar and Anjomshoa (2018) found that classification and regression tree (CART) analysis is useful for classifying patients into lower variability LoS groups. We applied CART analysis on the ICU LoS data from the partner hospital, and we classified patients into short-stay (SS), medium-stay (MS), and long-stay (LS) groups. In regression tree analysis, the dependent variable is recursively partitioned into groups according to the independent variables in a way that the fitted tree is capable of explaining the maximum variability in the dependent variable. In our case, the dependent variable was the LoS and the independent variables were patient attributes, such as primary procedure code, age, and gender that were known to us before surgery. According to the fitted regression tree, 40% of patients belong to the SS group, 51% belong to the MS group, and the remaining 9% belong to the LS group.

Next, we fitted Coxian discrete phase type (DPH) distributions (see Latouche & Ramaswami, 1999) to the data corresponding to each resource user group separately. We selected the model using the Akaike information criterion (AIC) and the Bayesian information criterion (BIC), and we validated the fit using the Chi-square goodness of fit test (see Hogg, Tanis, and Dale, 2015). When the AIC and BIC proposed different models, we selected the model that satisfied the goodness of fit test and had fewer parameters. The log-likelihood, the AIC and BIC values, and the Chi-square goodness of fit test's p-values of the fitted distributions are given in Table 1. We used fitted distribution models to sample patients' LoSs.

The literature suggests that the optimal occupancy level for the ICU is around 70% to 75% (see Tierney, Hons, Conroy, & Hons, 2014). In the hospital, seven ICU beds are being used for elective surgery patients. In our model, we place the restriction that a maximum of twenty patients, whose average LoS is 2.4 days, can be scheduled per week. This allows a utilisation of up to 97.9% of ICU bed days available and we refer to this (bed days utilised $\times$ 100/ bed days available) as the occupancy level of the ICU. However, the model decides the optimal throughput according to the desired trade-off (implemented as a penalty factor) between the throughput and the number of cancellations. We also constrain our model to maintain the proportion of the scheduled patients from each LoS group the same as the average proportion in the data. Furthermore, we refer to a four week period as one scheduling period, or one timeblock. A LoS scenario realisation is defined as an assignment of patients' positions in the queue and their LoS values in each group for a selected number of timeblocks. For example, an m-block LoS scenario realisation of n patients consists of a $n \times m$ matrix of LoS values, where n is the maximum number of patients that can be scheduled in one scheduling period, and each entry of the matrix is sampled from the fitted DPH distributions, independently of the others. An element, $LoS_{t,\,p}$, of the matrix represents the *LoS* of the p^{th} patient in timeblock t, $\forall\ p = 1, \ldots, n$, and $\forall\ t = 1, \ldots, m$.

While developing an MSS, we need to decide how many patients from each LoS group we should operate on, on each day of the scheduling period, to ensure smooth patient flow and

Table 1
The AIC and BIC, and the *Chi*-square test's *p*-value of the fitted DPH distributions.

Group	#Phase	Loglikelihood	# Param.	AIC val	BIC val	*p*-value
SS	3	−975.29	5	1960.59	1983.89	0.1173
MS	3	−1629.91	5	3269.82	3294.36	0.0917
LS	3	−483.13	5	976.26	992.22	0.5519

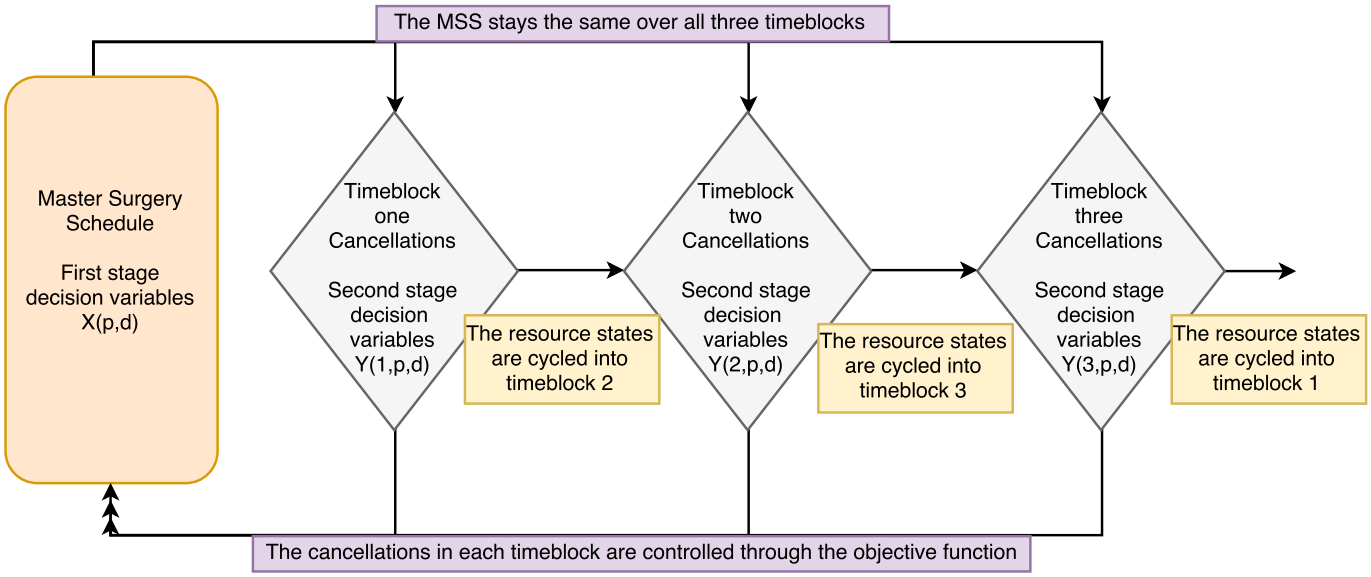


Fig. 1. Diagram representing the MSS being repeated over a three timeblock planning horizon and the interaction between timeblocks through cancellations and LoS rotation.

optimal throughput. We should make these decisions based entirely on patient groups' LoS distributions. Given an actual realisation of one-block LoS scenario as input, the model would take advantage of known LoS values and schedule particular subsets of patients from each group to balance resource requirements on each day. This is advantageous if we know the exact LoS values because we can manage patient flow more efficiently. However, we know only the LoS distributions. We developed a strategy to minimise the undesirable effect of providing LoS scenario realisation as input.

First, we modelled and optimised the patient flow process over several timeblocks and constrained our model to make the same scheduling decision for all the patients at position p in each timeblock, irrespective of their LoSs. Doing so helped us in optimising scheduling decisions over several possible LoS values, and the model could not take full advantage of a specific timeblock's realised LoSs. However, we obtained a schedule in which fewer patients were scheduled. This happened because the model had to ensure resource (ICU beds) availability for all the scheduled patients, irrespective of the variation in their resource requirements in different timeblocks. As a result, the model had to keep resources idle in some timeblocks. For example, consider a one bed ICU and a three-block time horizon. Assume that the patient at position one has a LoS of two, seven, and nineteen days in timeblock one, two, and three, respectively. If the model schedules this patient on the first day, it cannot schedule any patient until the nineteenth day in any timeblock because the scheduling decisions stay the same in each timeblock. Therefore, the bed-days utilised in timeblock one and two will be much less than the bed-days available, and our resource management will be inefficient.

To eliminate the undesirable effect of this feature of the model, we used a stochastic approach in which reverse decisions were modelled as cancellation decisions using decision variables ($Y_{t,\,p,\,d}$) that could cancel some patients' operations according to the capacity shortage in each timeblock, independently. In operational planning, cancelled patients should be rescheduled in the next available slots with some priority. Since we were developing a tactical MSS, we only identified patient groups, not patients. As a result, the MSS would be unaffected if we rescheduled a cancelled patient instead of a new patient in the next available slot for his or her group because each patient's LoS was randomly sampled from the fitted LoS distribution. In operational planning, we recommend that the cancelled patients are rescheduled early in the day to avoid cancelling them again. Now, we were able to generate efficient schedules. However, there was still some scope for the model to schedule particular subsets of patients on each day to balance resource requirements. As we increase the number of timeblocks, allowing the model to select patients in the MSS loses importance. However, the model could take advantage of the known LoS values while making cancellation decisions.

Next, we devised a strategy to prevent the model from taking advantage of this flexibility. We constrained our model to schedule patients in the same order as their positions in the queue in each group. As a result, the model could only decide the number of patients to be scheduled on each day of the scheduling period. For example, if the model decides to schedule four patients on the first week's Monday, then they have to be the first four patients, that is, patients at positions from one to four. Since the patients were positioned in the queue in random order, the model has to schedule a randomly-gathered patient subset on each day. This prevented the model from selecting particular subsets of patients to balance resource demand. We also constrained our model to cancel the earliest scheduled patient first on each day in each group. This prevented the model from identifying patients by taking advantage of known LoS values while making cancellation decisions. By using these strategies, we developed a novel approach to optimise patient flow by using a deterministic optimisation on different sequences of randomly generated LoS scenario realisations.

Table 2
An optimal MSS generated by the model for a three-block LoS scenario realisation.

Weekday	Mon	Tues	Wed	Thurs	Friday
timeblock one					
1^{st} week	p_1^4, p_2^6, p_3^2	p_4^5, p_5^3	p_6^2	p_7^5, p_8^2	$p_9^3, p_{10}^2, p_{11}^3$
2^{nd} week	$p_{12}^5, p_{13}^7, p_{14}^2, p_{15}^2$	$p_{16}^4 p_{17}^2$	p_{18}^2, p_{19}^8	p_{20}^{11}, p_{21}^2	p_{22}^3
3^{rd} week	$p_{23}^2, p_{24}^2, p_{25}^2, p_{26}^6$	~~p_{27}^2~~, p_{28}^3	~~p_{29}^3~~, $p_{30}^4, p_{31}^{10}, p_{32}^2$		$p_{33}^3, p_{34}^2, p_{35}^2$
4^{th} week	$p_{36}^4, p_{37}^2, p_{38}^4, p_{39}^9$	p_{40}^4	p_{41}^4, p_{42}^2		$p_{43}^4, p_{44}^3, p_{45}^3$
timeblock two					
1^{st} week	p_1^2, p_2^3, p_3^2	p_4^2, p_5^2	p_6^3	p_7^3, p_8^4	$p_9^2, p_{10}^{12}, p_{11}^4$
2^{nd} week	$p_{12}^7, p_{13}^2, p_{14}^3, p_{15}^2$	$p_{16}^2 p_{17}^2$	p_{18}^3, p_{19}^3	p_{20}^2, p_{21}^5	p_{22}^2
3^{rd} week	$p_{23}^2, p_{24}^2, p_{25}^2, p_{26}^{11}$	p_{27}^2, p_{28}^2	$p_{29}^2, p_{30}^3, p_{31}^{19}, p_{32}^3$		$p_{33}^2, p_{34}^2, p_{35}^2$
4^{th} week	$p_{36}^3, p_{37}^2, p_{38}^4, p_{39}^2$	p_{40}^{10}	p_{41}^8, p_{42}^5		$p_{43}^{11}, p_{44}^2, p_{45}^2$
timeblock three					
1^{st} week	p_1^2, p_2^3, p_3^4	~~p_4^5~~, ~~p_5^2~~	p_6^2	p_7^6, p_8^2	$p_9^2, p_{10}^2, p_{11}^2$
2^{nd} week	$p_{12}^2, p_{13}^2, p_{14}^4, p_{15}^3$	$p_{16}^6 p_{17}^2$	p_{18}^3, p_{19}^7	p_{20}^2, p_{21}^2	p_{22}^2
3^{rd} week	$p_{23}^4, p_{24}^2, p_{25}^2, p_{26}^2$	p_{27}^2, p_{28}^2	$p_{29}^3, p_{30}^3, p_{31}^9, p_{32}^2$		$p_{33}^4, p_{34}^2, p_{35}^3$
4^{th} week	$p_{36}^2, p_{37}^2, p_{38}^2, p_{39}^2$	p_{40}^6	p_{41}^2, p_{42}^2		$p_{43}^2, p_{44}^2, p_{45}^2$

Table 3
The MSS obtained for ten 130-block scenario realisations.

Day	1	2	3	4	5	8	9	10	11	12	15	16	17	18	19	22	23	24	25	26	T	C	O
SS_1	2	1	3	1	0	2	1	2	1	0	2	1	2	1	0	2	1	3	1	0	26		
MS_1	3	2	1	0	3	3	2	1	0	3	3	2	1	0	3	2	2	1	0	3	35		
LS_1	0	0	0	1	1	0	0	0	1	0	0	0	0	1	1	0	0	0	1	0	6		
S_1	5	3	4	2	4	5	3	3	2	3	5	3	3	2	4	4	3	4	2	3	67	6.5	76.2
SS_2	4	0	1	1	0	4	0	1	1	0	4	0	2	1	0	4	0	1	1	0	25		
MS_2	1	3	0	1	3	1	3	0	1	3	1	3	0	1	3	1	3	1	1	3	33		
LS_2	0	0	0	1	0	0	0	0	1	1	0	0	0	1	0	0	0	0	1	1	6		
S_2	5	3	1	3	3	5	3	1	3	4	5	3	2	3	3	5	3	2	3	4	64	5.6	74.2
SS_3	1	3	2	1	0	0	3	2	1	0	1	3	2	1	0	1	3	2	1	0	27		
MS_3	4	0	1	1	3	4	0	1	1	2	4	0	1	1	3	4	0	1	1	3	35		
LS_3	0	0	0	1	0	0	0	0	1	1	0	0	0	1	0	0	0	0	1	1	6		
S_3	5	3	3	3	3	4	3	3	3	3	5	3	3	3	3	5	3	3	3	4	68	6.4	77.1
SS_4	1	3	1	3	0	1	3	1	2	0	1	3	1	2	0	1	3	1	2	0	29		
MS_4	4	0	2	0	3	4	0	2	1	3	4	0	2	1	3	4	0	2	1	3	39		
LS_4	0	0	0	1	1	0	0	0	1	0	0	0	0	1	1	0	0	0	1	0	6		
S_4	5	3	3	4	4	5	3	3	4	3	5	3	3	4	4	5	3	3	4	3	74	10.8	81
SS_5	2	2	1	1	1	1	2	1	1	1	1	2	1	1	1	1	2	1	1	1	25		
MS_5	3	0	2	1	2	3	1	2	1	2	3	1	2	1	2	3	0	2	1	2	34		
LS_5	0	0	0	1	1	0	0	0	0	1	0	0	0	1	1	0	0	0	0	1	6		
S_5	5	2	3	3	4	4	3	3	2	4	4	3	3	3	4	4	2	3	2	4	65	7.4	77.9
SS_6	3	2	2	0	0	3	2	2	0	0	3	2	2	0	0	3	2	1	0	0	27		
MS_6	2	1	2	1	2	2	1	2	1	3	2	1	2	1	3	2	1	2	1	3	35		
LS_6	0	0	0	1	1	0	0	0	0	1	0	0	0	1	1	0	0	0	0	1	6		
S_6	5	3	4	2	3	5	3	4	1	4	5	3	4	2	4	5	3	3	1	4	68	8.2	77.3
SS_7	1	0	1	3	0	1	0	2	3	0	1	0	2	3	0	1	0	2	3	0	23		
MS_7	3	2	0	0	3	3	2	0	0	3	3	1	0	0	3	3	2	0	0	3	31		
LS_7	0	0	0	1	1	1	0	0	0	0	0	0	0	1	1	1	0	0	0	0	6		
S_7	4	2	1	4	4	5	2	2	3	3	4	1	2	4	4	5	2	2	3	3	60	5.5	71.9
SS_8	4	2	0	0	0	4	2	0	0	0	4	2	0	0	0	4	2	1	0	0	25		
MS_8	1	0	2	3	2	1	0	2	4	2	1	0	2	3	2	1	0	2	3	2	33		
LS_8	0	0	0	1	1	0	0	0	0	1	0	0	0	1	1	0	0	0	0	1	6		
S_8	5	2	2	4	3	5	2	2	4	3	5	2	2	4	3	5	2	3	3	3	64	6.1	75.7
SS_9	4	0	2	0	0	4	0	2	0	0	4	1	2	0	0	4	1	2	0	0	26		
MS_9	0	3	0	3	3	0	3	0	2	3	0	3	0	3	3	0	3	0	2	3	34		
LS_9	0	0	0	1	0	0	0	1	1	0	0	0	0	1	0	0	0	1	1	0	6		
S_9	4	3	2	4	3	4	3	3	3	3	4	4	2	4	3	4	4	3	3	3	66	5.9	75.8
SS_{10}	4	0	1	0	0	4	0	2	0	0	4	0	1	0	0	4	0	2	0	0	22		
MS_{10}	0	3	1	2	2	0	3	1	2	2	0	3	1	2	3	0	3	1	2	2	33		
LS_{10}	0	0	0	1	0	0	0	0	1	1	0	0	0	1	0	0	0	0	1	1	6		
S_{10}	4	3	2	3	2	4	3	3	3	3	4	3	2	3	3	4	3	3	3	3	61	4	73.4
SAA	4.7	2.7	2.5	3.2	3.3	4.6	2.8	2.7	2.8	3.3	4.6	2.8	2.6	3.2	3.5	4.6	2.8	2.9	2.7	3.4	65.7		
SAA_R	5	3	3	3	3	5	3	3	3	3	5	3	3	3	4	5	3	3	3	3	66		

Our stochastic model uses the LoS scenarios in a non-traditional manner. In the traditional approach, the first stage variables represent the decisions we make before any uncertainty is revealed, and the second stage variables represent the decisions we make according to each realised scenario. In our model, first stage variables are associated with the master plan that repeats over time for the whole planning horizon, whereas the second stage variables are cancellation decisions that occur sequentially over time. Moreover, since we extend patients' remaining LoS from one scheduling period to the next scheduling period, each timeblock's resource states (the bed occupancy) interact with other timeblocks' resource states over the whole planning horizon. Fig. 1 represents how we used LoS scenarios to model uncertainty in the patient flow process in a novel way.

Here, $X_{p,\,d}$ are binary decision variables, associated with the master plan that take the value one if patients at position p in the queue are scheduled for surgery on day d of each timeblock; zero otherwise, and $Y_{t,\,p,\,d}$ are cancellation decision variables that take the value one if the patients at position p in timeblock t is cancelled on day d. The values of $X_{p,\,d}$ stay the same for all the patients at position p in all timeblocks, whereas $Y_{t,\,p,\,d}$ can take different values as t varies. A typical schedule generated by using the full model for an input of three-block LoS scenario realisation is given in Table 2.

In this table, the ps' subscripts represent patients' positions in the queue and the superscripts represent their LoSs. The table shows which patients should be operated on, on each day, and the struck out patients are the ones whose operations are cancelled because of a lack of resources. Patients at a particular position in the queue are scheduled at the same time in each timeblock irrespective of their LoSs. For example, patients at position 31 are operated on, on the third week's Wednesday in each timeblock whereas their LoSs are ten, nineteen, and nine. However, some patients' operations may get cancelled according to resource unavailability. For example, in the first timeblock, patients at position 27 and 29 are cancelled.

5. The model

In this section, we will first define all the notation and then present the MIP model. The parameter g represents the number of patient groups and n represents the maximum number of patients allowed to be scheduled in each timeblock. We use m to denote the number of timeblocks considered in the planning horizon. The parameter *beds* refers to the number of beds (resources) available in the ICU. Since all the patients in a group are the same from a scheduling perspective, we refer to them by their positions in the queue in each timeblock. The set $Patients = \{1, 2, \ldots, n\}$, denotes the set of all patients in each timeblock and the set $timeblocks = \{1, 2, \ldots, m\}$, contains all the scheduling periods in the planning horizon. Similarly, the set $Group_k$ refers to the set of patients in group k, $\forall k = 1, \ldots, g$. Finally, the set of all days, working days, and weekends in each scheduling period are denoted by *days*, *weekdays*, and *weekends*, respectively. A detailed table of all the notation used is given in the appendix.

5.1. Objective function

Our objective is to maximise a weighted throughput while maintaining a balance between resource utilisation and cancellations. We measure resource utilisation and cancellations by summing the LoSs of patients in the MSS and the LoSs of cancelled patients, respectively. A multiplier f is used to enforce a desirable balance between these two conflicting objectives. The objective of the optimisation is to maximise

$$obj = \sum_{t \in timeblocks} \sum_{p \in Patients} LoS_{t,p} \times \sum_{d \in days} (X_{p,d} - f \times Y_{t,p,d}). \tag{1}$$

5.2. Set of constraints

As we discussed in the Introduction, there are some unique features of the elective patient flow process, and we captured these features in our model by using several constraints as follows.

5.2.1. Resource availability constraints

In hospitals, elective patients are operated on every weekday and they stay in the ICU or a surgical ward until they recover. The long-stay patients or the patients who are operated on at the end of a scheduling period may continue their stay in the ICU or a surgical ward in the next scheduling period. As a result of this, at the beginning of each scheduling period, some resources are occupied by the patients from earlier scheduling periods. This means, that not all the resources are available for allocation on the first day of a scheduling period. We modelled this feature by making use of a rotation schedule such as those used in crop rotation planning (see Santos, Munari, Costa, & Santos, 2015). For example, if we solve a model for thirteen scheduling periods, then a patient who was operated on in the last scheduling period will stay in the facility during that period. However, if that patient's LoS is more than the remaining days in the period, then we rotated that patient's stay into the first scheduling period. While ensuring resource availability for patients in each timeblock, we take into account patients from the current timeblock and the previous two timeblocks, and the timeblocks are numbered in cyclic order. For example,

$$\sum_{p \in Patients} \Bigg\{ \sum_{j=0}^{LoS_{1,p}-1} (X_{p,d-j} - Y_{1,p,d-j}) + \sum_{j=0}^{LoS_{m,p}-1} (X_{p,28+d-j} - Y_{m,p,28+d-j}) + \sum_{j=0}^{LoS_{m-1,p}-1} (X_{p,56+d-j} - Y_{m-1,p,56+d-j}) \Bigg\} \leq beds, \quad \forall d \in days, \tag{2}$$

are the required constraints for scheduling period one. They state that the number of patients who were scheduled in periods 1, m, and $m-1$, and were staying in the ICU on day d of scheduling period one, must be less than or equal to the number of beds (resources) in the ICU. The required constraint for scheduling period two is

$$\sum_{p \in Patients} \Bigg\{ \sum_{j=0}^{LoS_{2,p}-1} X_{p,d-j} - Y_{2,p,d-j} + \sum_{j=0}^{LoS_{1,p}-1} X_{p,28+d-j} - Y_{1,p,28+d-j} + \sum_{j=0}^{LoS_{m,p}-1} X_{p,56+d-j} - Y_{m,p,56+d-j} \leq beds \Bigg\} \leq beds, \quad \forall d \in days. \tag{3}$$

Similarly, the patients who were operated on in scheduling periods $t-1$ and $t-2$ might stay in the ICU during period t, $\forall t = 3, \ldots, m$, and the required constraints to make sure that the number of patients on each day d of timeblock t is less than or equal to the number of beds are

$$\sum_{p \in Patients} \Bigg\{ \sum_{j=0}^{LoS_{t,p}-1} X_{p,d-j} - Y_{t,p,d-j}$$

$$+ \sum_{j=0}^{LoS_{t-1,p}-1} X_{p,28+d-j} - Y_{t-1,p,28+d-j} + \sum_{j=0}^{LoS_{t-2,p}-1} X_{p,56+d-j} - Y_{t-2,p,56+d-j} \leq beds \Bigg\} \leq beds, \quad \forall d \in days. \tag{4}$$

Note that, we considered the two previous periods in these constraints. This is because the LoS data we used allows patients to stay in the ICU for up to three timeblocks. This could be easily modified to take into account longer or shorter LoSs. Rotational scheduling is helpful in modelling the actual process in which not all the resources are available at the beginning of the scheduling period. Furthermore, rotation scheduling also helps us avoid some infeasible and impractical schedules. For example, when we group patients according to their predicted LoS and maximise the weighted throughput without considering rotation, the model tends to schedule all long stay patients towards the end of the scheduling period. By doing so, the model can ignore patients' remaining LoS. Rotation scheduling removes this unrealistic incentive.

5.2.2. Constraints to avoid multiple scheduled arrivals of the same patient

Each patient can be scheduled for an operation at most once. The required constraints are

$$\sum_{d \in weekdays} X_{p,d} <= 1, \quad \forall p \in Patients. \tag{5}$$

5.2.3. Constraints to avoid scheduling patients' operations over weekends

The constraints,

$$\sum_{d \in weekends} X_{p,d} <= 0, \quad \forall p \in Patients, \tag{6}$$

make sure that no patient is scheduled over weekends.

5.2.4. Constraints to make sure that only scheduled patients can get cancelled

We cannot cancel a patient's operation unless it is scheduled. The constraints to avoid cancelling unscheduled patients are

$$Y_{t,p,d} \leq X_{p,d}, \quad \forall t \in timeblocks, \quad \forall p \in Patients, \quad \forall d \in days. \tag{7}$$

5.2.5. Constraints to impose first come first served policy

As we discussed in Section 4, we need to constrain our model to schedule patients in the same order as their position in the queue in each group, and we refer to these constraints as ordering constraints. However, the model is free to schedule patients from different groups optimally, irrespective of their positions. For example, a long-stay patient at position eight can be scheduled before a short-stay patient at position two. As we will discuss further in Section 6, by using these constraints we were also able to obtain a faster convergence to optimality, and a better estimation of the number of cancellations and occupancy levels. The required inequalities for the ordering constraints are

$$X_{p,d} \leq \sum_{t=1}^{d} X_{p-1,t}, \quad \forall \{p, p-1\} \in Group_s, \quad \forall d \in days, \quad \forall s = 1..g, \tag{8}$$

and

$$p \times X_{p,d} \leq k_s + \sum_{i \in Group_s} \sum_{t=1}^{d-1} X_{i,t}, \quad \forall p \in Group_s, \quad \forall d \in days, \quad \forall s = 1..g. \tag{9}$$

Here, the constraints in (9) are redundant. However, they are valid inequalities and proved effective in reducing convergence times.

5.2.6. Constraints to impose first come first cancel policy

As we discussed in Section 4, to prevent the model from exploiting the knowledge of known LoS values while making cancellation decisions, we need to enforce a consistent cancellation order. Therefore, we constrained our model to cancel the earliest scheduled patient first on each day in each group. The required constraints for patient group s are

$$Y_{t,p,d} + 1 \geq X_{p,d} + Y_{t,p+1,d}, \quad \forall s = 1..g, \quad \forall p \in Group_s, \quad \forall d \in weekdays, \quad \forall t \in timeblocks. \tag{10}$$

Furthermore, we may add a few more constraints if we need to ensure that a minimum number of patients from each group is scheduled each week or the maximum allowable cancellations over the planning horizon is less than some percentage of the number of scheduled patients. Similarly, we may add some constraints to make sure that patients from each group are scheduled for operations in accordance with surgeons' availability. Adding these extra constraints can change our tactical decision model to a more operational tool where other decisions such as surgeons' rosters need to be considered.

6. Computational experiments and results

In this section, we describe various experiments to evaluate the features of our model. First, we used all the three LoS groups, that is, the short-stay (SS) group, the medium-stay (MS) group, and the long-stay (LS) group, and we developed a robust MSS for the hospital. We defined the robust MSS as the schedule that performed better than the other schedules when all the schedules generated by using various LoS scenario realisations were evaluated on an independent very-long LoS scenario realisation. The last four experiments are designed to help us understand various features of the model.

6.1. A robust MSS to optimise the surgical patient flow in the ICU

In the first experiment, we optimised the patient flow for ten 130-block LoS scenario realisations of all the three LoS groups, and we obtained ten different MSSs. As discussed in Section 4, we sampled the LoS scenario realisations from the fitted DPH distribution of each LoS group, and we used the full model with the objective given in Eq. (1) and constraints (2) to (10) to develop schedules for ten 130-block LoS scenario realisations. We also added some constraints to limit the variation in the number patients scheduled on each weekday of the scheduling period. For example, the difference between the number of the SS patients scheduled on the first and the second Monday of the scheduling period cannot be more than one. These constraints helped us reduce the search space, and obtain a feasible solution quickly. We set the solver time limit to 72 hours, and the obtained schedules had optimality gaps less than ten percent.

The obtained schedules are given in Table 3. In the table, each box of four rows represents a schedule for a LoS scenario realisation. There are twenty working days in a scheduling period. The topmost row represents the day number. The leftmost column displays the schedule name according to the patient group. The first three rows of each schedule tell us how many patients from each group are scheduled, and the last row indicates the total number of patients being scheduled, on each day of a timeblock. Column T shows the total number of patients being scheduled per timeblock in each LoS group. Average number of cancellations per

Table 4
The performance of obtained schedules when we evaluated them using simulation.

Schedule	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}
$Canc_{eval}$	7	5.3	6.9	10.5	5.8	7.4	4.3	5.1	5.6	3.7
$Occu_{eval}$	75.7	73.6	76.6	79.5	74.8	76.1	70.5	73.7	75	72.4
Canc	6.5	5.6	6.4	10.8	7.4	8.2	5.5	6.1	5.9	4.0
Occu	76.2	74.2	77.1	81.0	77.9	77.3	71.9	75.7	75.8	73.4

timeblock and the occupancy levels are given in columns *C* and *O*, respectively.

Table 3 shows that all the obtained schedules are different, and the model finds the best fitted schedule for each LoS scenario realisation. Although all the schedules are different, they are qualitatively consistent. For example, the highest number of patients (four or five patients) are scheduled on each week's Monday in almost all the schedules. In most of the schedules, SS patients are scheduled early in the week whereas MS patients are scheduled in the middle and the end of the week. Moreover, the rounded sampling average approximation (SAA) of the total number of patients scheduled on each day given in the second last row is almost periodic with a period of one week.

Next, we evaluated the performance of all the schedules on an independent 1300-block LoS scenario realisation by using a simulation model. The average occupancy level and the average number of cancellations per timeblock computed by the simulation model are given in Table 4. For an easy comparison, we also list the average occupancy level and the average number of cancellations computed by the MIP model, in the last two rows.

The average number of cancellations and the occupancy level indicate that all the schedules performed well when tested on an independent LoS scenario realisation, and there is a trade-off between the average number of cancellations and the number of patients scheduled. For example, S_4 has a very high number of scheduled operations (74 per timeblock) and cancellations (10.5 per timeblock). On the other hand, S_7 and S_{10} have less scheduled operations and fewer cancellations. The remaining schedules are somewhere in between. Although all the schedules performed well, some of them performed better than the others. For example, S_9 dominates S_1, S_2, S_3, S_5, S_6, and S_8, because either S_9 has more scheduled operations in comparison to the others for a similar number of cancellations, or it has less cancellations for a similar throughput. Similarly, S_{10} dominates S_7. From this analysis, we conclude that S_9 is the robust schedule among S_1, S_2, S_3, S_5, S_6, S_8, and S_9, and the others are not comparable because of completely different numbers of scheduled operations and cancellations.

Most of the schedules we obtained were slightly different from each other because of three main reasons. First, a majority of patients stayed in the ICU for only one day. Therefore, shifting a patient's operation to the next day or the previous day according to the LoS scenario realisation generated a new schedule without significant changes in the objective function's value. Second, SS and MS patient groups did not differ much in terms of the average LoS. Although the model was able to exploit this difference to obtain slightly better performing schedules, the various possible rearrangements of these groups increased the combinatorial aspect of the problem and the number of alternative efficient schedules. Finally, the four-week scheduling period also increased the number of equally efficient alternative schedules.

In the next four experiments, we evaluate various features of the model. In order to demonstrate the usefulness of various features of the model distinctly, we used a new patient group obtained by merging SS and MS patients, and we increased each patient's LoS by one day. From here onward, we refer to the new patient group with the modified LoS as the SS group and the LS group remains the same. In order to simplify the experiments, we used only the SS group in the next three experiments, and we allowed a maximum of 52 patients could be scheduled in each timeblock. There were only three out of 30 LoS scenario realisations for which this maximum was attained.

Table 5
Optimal *MSS*s generated by the full model for thirteen-block scenario realisations.

Schedule identity	Mon	Tues	Wed	Thurs	Friday	Sum	Freq
MSS_1	4	1	3	1	3	12	11
MSS_2	4	1	2	2	3	12	3
MSS_3	4	1	3	2	3	13	0
MSS_4	4	1	2	1	3	11	5
MSS_5	4	2	2	1	3	12	3
MSS_6	4	1	2	2	2	11	2
MSS_7	4	2	2	2	3	13	2
MSS_8	4	1	3	1	4	13	1
MSS_9	4	1	2	1	4	12	1
MSS_{10}	3	3	1	2	3	12	1
MSS_{11}	3	2	2	1	3	11	1
MSS_{12}	4	2	1	2	3	12	0
SAA	3.93	1.27	2.37	1.27	3.00	11.84	NA
Mode	4	1	2	1	3	11	NA
MSS_R	4	1	3	1	3	12	NA
$MSS_{R_{f=4}}$	4	1	2	1	3	11	NA

6.2. Experiment to enhance the robustness of the MSS

Since we obtained very competitive schedules for various LoS scenario realisations in the previous experiment, we performed an experiment to distinctly recognise the robust schedule. We reduced the number of independent decisions made in the master plan by imposing weekly periodicity. Periodicity is a desirable feature and it makes the MSS more practical for elective surgery planning in hospitals. We constrained our model to schedule the same number of patients on the same weekday. The resultant optimal schedules for 30 thirteen-block LoS scenario realisations are given in Table 5. Each row indicates an optimal MSS obtained for one or more LoS scenario realisations, and the value in the rightmost column is the number of instances for which that schedule is the optimal schedule. There are also two MSSs (MSS_3 and MSS_{12}) which were not the optimal schedule for any LoS scenario realisation in this experiment. However, they were the optimal MSSs in other experiments. We use the same notation to refer to these MSSs in all experiments as given in Table 5.

Table 5 shows that MSS_1 is the most frequent optimal schedule. Still maintaining our periodicity constraint, we also obtained optimal schedules for five 104-block (eight years) LoS scenario realisations to compare it with the others. Four out of five LoS scenario realisations generated MSS_1 and one generated MSS_2, confirming our hypothesis that the MSS becomes more robust with respect to LoS scenario realisations as we increase the planning horizon. This happened because of the fact that increasing the planning horizon resulted in our scheduling decisions being optimised over a larger number of LoS scenario realisations. Moreover, MSS_1 also performed very well for the 104-block LoS scenario realisation that generated MSS_2 as the optimal schedule. For example, there were only two more cancellations and 40 less bed-days used

Table 6
Optimal *MSSs* generated by the relaxed model for thirteen-block scenario realisations.

Schedule identity	Mon	Tues	Wed	Thurs	Friday	Sum	Freq
MSS_1	4	1	3	1	3	12	10
MSS_2	4	1	2	2	3	12	5
MSS_3	4	1	3	2	3	13	10
MSS_8	4	1	3	1	4	13	5
SAA	4	1	2.8	1.5	3.2	12.5	NA

Table 7
Frequency of MSSs generated by the full model for longer planning horizons.

Timeblocks	26-blocks (30 realisations)		104-blocks (five realisations)	
Schedule Id	Frequency	Proportion	Frequency	Proportion
MSS_1	14	0.47	4	0.80
MSS_2	6	0.20	1	0.20
MSS_3	2	0.07	0	0.00
MSS_4	3	0.10	0	0.00
MSS_6	1	0.03	0	0.00
MSS_7	3	0.10	0	0.00
MSS_{12}	1	0.03	0	0.00

Table 8
The robust MSS for generated by the model using a 104-block LoS scenario realisation which includes both: long stay and short stay patients.

Number of patients to be operated on each week of the schedule					
Day of the week	Monday	Tuesday	Wednesday	Thursday	Friday
Short stay patients	4	1	2	0	3
Long stay patients	0	0	0	1	0
Total patients	4	1	2	1	3

in eight years when we imposed MSS_1 on that LoS scenario realisation. We will refer to MSS_1 as MSS_R (robust MSS). We can see from Table 5 that the SAA is not exactly the same as MSS_R. In fact, MSS_R is the same as the most frequent optimal schedule for thirteen-block LoS scenario realisations. We concluded from the experiment that as we increase the planning horizon, the MSS becomes more robust.

6.3. Experiment to assess the effectiveness of the ordering constraints

Next, we repeated the same experiment after removing the ordering constraints in Sections 5.2.5 and 5.2.6. The purpose of this experiment was to understand the effect of these constraints. The MSSs obtained for the same 30 thirteen-block LoS scenario realisations are given in Table 6. We obtained four possible MSSs in this case and there were two most frequent optimal schedules (MSS_1 and MSS_3) each with a frequency of ten. We refer to the model with the ordering constraints as the full model and the model without the ordering constraints as the relaxed model.

The relaxed model scheduled 52 patients per timeblock for half of the LoS scenario realisations, whereas the full model did so for only three (out of 30) LoS scenario realisations. If we allowed the model to schedule more than 52 patients each timeblock, it could do so only for some rare LoS scenario realisations. This is because weekly periodicity required the model to increase the number of patients in each timeblock by at least four and the hard constraint on the number of resources available did not permit scheduling 56 patients each timeblock. Since the relaxed model had the flexibility to schedule particular subsets of patients, it was able to schedule 52 patients per timeblock by gathering similar resource users into patient subsets and then scheduling these subsets in such a way that it can achieve maximum utilisation of available resources. We discuss this behaviour thoroughly in the next two paragraphs.

Furthermore, we obtained a smaller number of optimal schedules by using the relaxed model as compared to the number of optimal schedules obtained by using the full model. This was a seemingly counter-intuitive result because by fixing the order of the selected patients in the MSS we would first expect that the model would not be able to take advantage of known LoS values and therefore the MSS should converge faster to the most frequent optimal schedule. We analysed detailed solutions generated by both models for one of the LoS scenario realisations. The full model generated MSS_2 and the relaxed model generated MSS_1 as the optimal schedule.

As described in Section 4, our thirteen-block LoS scenario realisation was a 13×52 matrix where the column numbers represented the patients' positions and the row numbers represented their timeblock. Although the LoS scenarios were sampled randomly from the fitted DPH distributions, there was a reasonably large sampling variability (the variation in sample means and sample variances) among columns of the matrix because of a small sample size (only thirteen). Since the relaxed model had the flexibility to schedule columns in any order, it exploited the knowledge of the columns' sampling variances while developing an optimal schedule. We observed that the model tended to schedule columns with small sample variances on Monday, Tuesday, and Friday, whereas the columns with large sample variances were scheduled on Wednesday and Thursday. One intuitive explanation for this behaviour is that as our data was positively skewed, columns with large sample variance included relatively longer LoS patients. Because of the formulation of the objective function, the model tended to favour longer LoS patients and it avoided cancelling them. Longer LoS patients also improved resource utilisation over weekends. The model scheduled them in the middle of the week so that it could cancel shorter LoS stay patients at the end of the week if the need arose. Similarly, it scheduled columns with small variability at the beginning of the week to ensure resource availability for longer stay patients later on.

The full model had to schedule columns in order, and it relied on adjusting the number of patients being scheduled on each weekday and the position of the patients who could be cancelled to obtain a better schedule for each LoS scenario realisation. For example, the detailed solution we analysed had MSS_2 as the optimal schedule and the patients at position 32 were cancelled in five out of thirteen timeblocks. Cancelling patients at position 32 without cancelling patients at positions 29 and 30 was not possible in MSS_1 because they were not the earliest scheduled patients. Therefore, the full model generated MSS_2 instead of MSS_1 as the optimal schedule. Although by allowing the order to be broken, the model seemed to be less affected by a particular sample path, it was an incorrect inference. The model should not have differentiated among columns of the LoS scenario realisations because they belong to the same patient group, and the ordering constraints helped us achieve that objective.

Furthermore, the larger variation observed in the schedules generated by the full model vanished when we increased the length of the sample path or the planning horizon. The claim is evident from the fact that the number of MSSs obtained reduces from eleven in Table 5 to seven in Table 7 with MSS_1 appearing 47% of the time when we increase the time horizon from thirteen-blocks to 26-blocks. Our claim is also supported by the results in Section 6.2 where we obtained only two MSSs with MSS_1 appearing 80% of the time when we increased the time horizon to 104-blocks.

We also developed a simulation model to calculate the actual number of cancellations and the true occupancy level achieved by implementing MSS_R and MSS_3 (MSS_R was the most frequent optimal schedule in Sections 6.2 and 6.3 and MSS_3 was another equally

Table 9
Comparison of the results (occupancy rates and average cancellations per timeblock) generated by the two models.

	Relaxed model (without the ordering constraints)				Full model (with the ordering constraints)			
Scenario realisation	Average cancellations	Total Scheduled	Cancellation (%)	Occupancy level (%)	Average cancellations	Total Scheduled	Cancellation (%)	Occupancy level (%)
LoS1	3.2	52	6.21	86.97	4.8	48	9.17	76.84
LoS2	3.6	52	6.95	89.05	4.8	48	9.17	78.96
LoS3	3.0	52	5.77	89.72	4.5	48	8.73	80.02
LoS4	1.6	48	3.11	86.81	7.8	52	14.94	83.59
LoS5	2.2	52	4.29	87.68	3.5	48	6.80	78.18
LoS6	2.2	48	4.14	88.97	3.2	44	6.21	78.14
LoS7	3.4	52	6.51	89.17	4.6	48	8.88	79.98
LoS8	1.0	48	1.92	86.66	4.9	48	9.47	79.95
LoS9	2.0	52	3.85	87.95	3.5	48	6.80	77.94
LoS10	3.4	52	6.51	89.36	4.5	48	8.73	80.14
LoS11	1.2	48	2.37	88.27	5.8	48	11.09	82.54
LoS12	1.8	48	3.40	87.13	2.8	44	5.47	76.77
LoS13	1.9	52	3.70	88.03	3.9	48	7.54	78.53
LoS14	2.7	52	5.18	88.70	3.8	48	7.25	78.41
LoS15	3.6	52	6.95	89.87	4.5	48	8.73	80.81
LoS16	3.5	52	6.66	89.05	4.8	48	9.32	80.38
LoS17	0.7	48	1.33	86.50	2.0	44	3.85	76.77
LoS18	1.7	48	3.25	88.93	2.7	44	5.18	78.73
LoS19	3.2	52	6.07	88.97	6.3	52	12.13	82.61
LoS20	1.2	48	2.22	86.07	5.1	48	9.76	79.79
LoS21	1.2	48	2.22	86.07	5.4	48	10.36	79.36
LoS22	2.3	48	4.44	89.52	3.6	44	6.95	77.59
LoS23	2.8	52	5.33	88.03	6.5	52	12.43	82.14
LoS24	1.8	52	3.55	87.24	3.8	48	7.25	77.04
LoS25	1.1	48	2.07	86.46	4.8	48	9.32	79.63
Simulation results when we implemented				MSS_R	4.31	48	8.98	78.47
				MSS_3	6.96	52	13.38	80.66

Table 10
Comparison of occupancy rates and average cancellations per timeblock for two penalty factors.

	f=2				f=4			
Scenario realisation	Average cancellations	Total Scheduled	Cancellation (%)	Occupancy level (%)	Average cancellations	Total Scheduled	Cancellation (%)	Occupancy level (%)
LoS1	5.1	48	10.58	77.00	1.0	40	2.08	68.17
LoS2	2.4	44	4.97	76.33	2.4	44	4.97	76.33
LoS3	4.7	48	9.78	80.57	1.6	44	3.37	75.94
LoS4	2.2	44	4.49	76.22	0.8	40	1.60	71.47
LoS5	3.5	48	7.21	78.65	0.8	40	1.76	69.15
LoS6	3.0	44	6.25	78.77	3.1	44	6.41	78.77
LoS7	5.0	48	10.42	80.30	1.3	40	2.72	71.31
LoS8	3.9	48	8.17	80.57	1.1	40	2.24	71.08
LoS9	3.9	48	8.17	78.61	2.1	44	4.33	73.94
LoS10	4.0	48	8.33	79.63	1.2	40	2.40	69.62
LoS11	5.2	48	10.90	81.95	2.2	44	4.65	77.51
LoS12	4.8	48	10.10	80.65	1.5	40	3.04	71.74
LoS13	4.0	48	8.33	78.53	1.4	44	2.88	74.22
LoS14	4.2	48	8.65	78.65	2.4	44	4.97	73.98
LoS15	4.8	48	9.94	80.65	1.3	40	2.72	69.70
LoS16	4.8	48	9.94	79.83	2.3	44	4.81	76.14
LoS17	1.8	44	3.85	76.18	1.8	44	3.85	76.18
LoS18	4.5	48	9.46	81.79	1.0	40	2.08	70.33
LoS19	4.5	48	9.46	78.73	1.1	40	2.24	69.86
LoS20	5.2	48	10.90	79.16	1.5	40	3.21	70.60
LoS21	5.1	48	10.58	79.40	1.2	40	2.56	69.82
LoS22	2.8	44	5.93	77.12	0.6	36	1.28	67.97
LoS23	4.2	48	8.81	79.40	2.1	44	4.33	75.67
LoS24	4.0	48	8.33	78.10	1.3	40	2.72	69.19
LoS25	4.2	48	8.81	81.20	0.5	40	0.96	70.92

frequent optimal schedule as MSS_R in Section 6.3). We found that the results from the relaxed model understated the number of cancellations and overstated the occupancy level in comparison to the results from the simulation model. A detailed comparison of the results is given in Table 9 in the appendix. As evident from Table 9, the results generated by the full model are quite close to the results generated by the simulation model. We can also see that MSS_3 performs slightly better than MSS_R in terms of occupancy level, whereas MSS_3's performance is much worse than that of MSS_R in terms of the number of cancellations. Keeping the number of cancellations within limits is an important goal of the hospital management. Misleading results from the relaxed model may entice us to implement MSS_3 and underachieve that goal.

Furthermore, the full model converged to optimality much faster than the relaxed model because of a drastic reduction in the search space. On average, a thirteen-block LoS scenario reali-

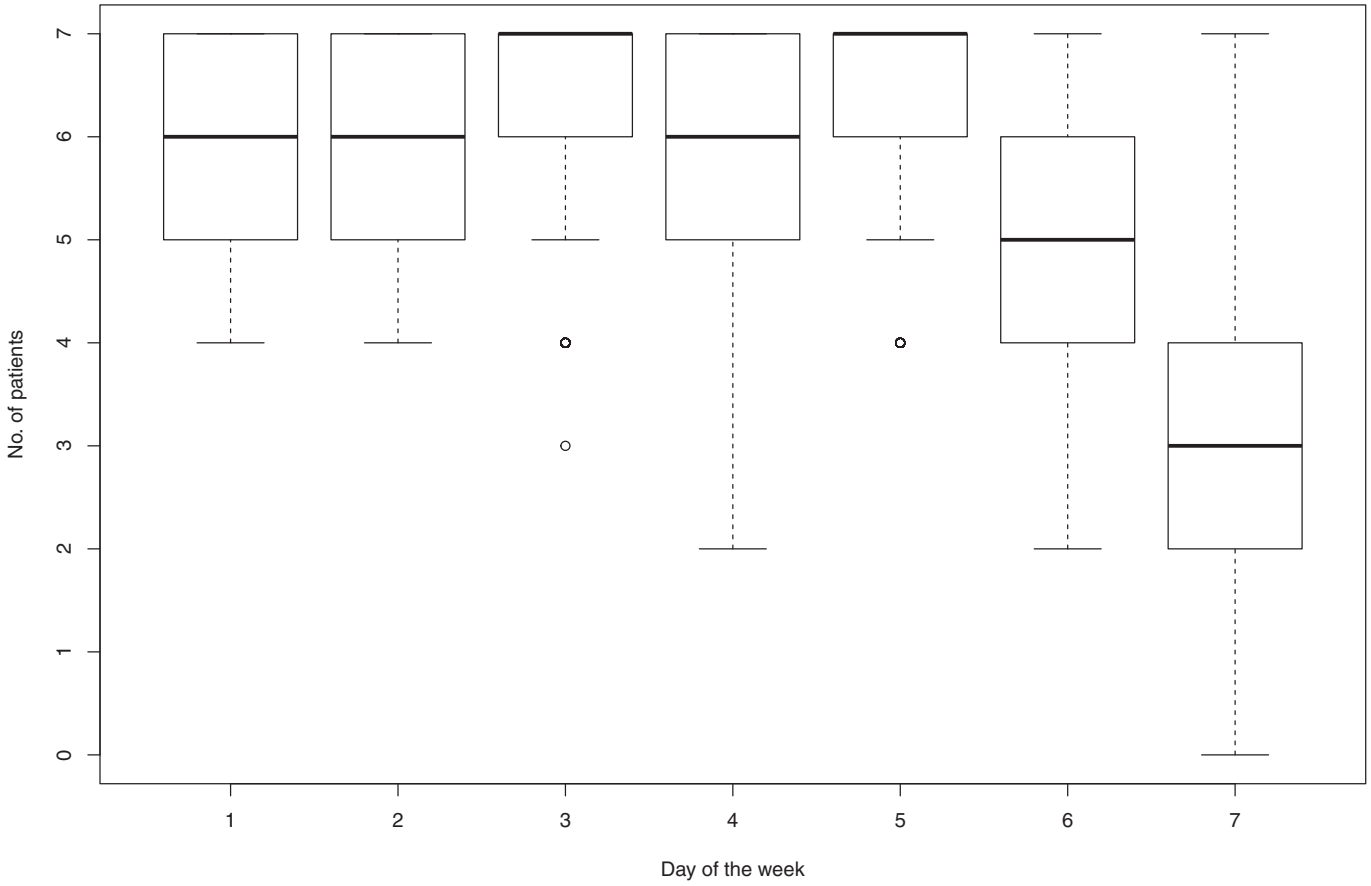


Fig. 2. Boxplot of the number of patients in the ICU on each weekday with MSS_1.

sation reached a 1% optimality gap in 120 minutes by using the relaxed model, whereas the same instance was solved to optimality within twenty minutes by the full model. From the analysis, we concluded that the ordering constraints helped us in limiting the model's ability to take advantage of known LoS values and gave us accurate estimation of occupancy level and cancellations, and they made the computation faster. We also concluded that classifying patients into lower variability LoS groups was useful to manage patient flow more efficiently. Moreover, we analysed the results from the simulation model and we provide the box-plots of the number of occupants in the ICU on each day of the week obtained by implementing MSS_R and MSS_3 in Figs. 2 and 3, respectively, in the appendix. Although there is variation in the number of patients scheduled on each weekday, it is evident from the box-plots that the variation in the ICU bed demand on each weekday is reasonably small.

6.4. Experiment to evaluate the usefulness of the cancellation trade-off parameter

Another important feature of our model was that we could decrease cancellations at the cost of a lower occupancy level, or improve the occupancy level which would increase cancellations. Therefore, in one of our experiments, we changed the control parameter, f, from two to four. A detailed comparison of occupancy level and cancellations at the two penalty levels is given in Table 10 in the appendix. The most frequently obtained schedule was the schedule $MSS_{R_{f=4}}$ in Table 5. This experiment indicated that our model is a flexible tool for decision makers to help them in making optimal decisions.

6.5. Analysing an MSS developed by using a reduced variability LoS distribution

Finally, we performed an experiment to understand the effect of reducing the variation in the LoS data, and we used both the SS and the LS patient groups in this experiment. When we use average LoS or make some simplifying assumptions regarding the LoS distributions, we change the variability in the LoS. In our case, since the average LoS for each patient group was not an integer, we modelled the LoS with a discrete random variable with only two possible values {3, 4} while keeping the average LoS the same. In order to compare the results, we also generated the optimal MSS for the ICU with random LoS scenario realisations, and the most frequently optimal schedule is given in Table 8.

We found that the MSS obtained was drastically different from the MSS in Table 8. For example, five short stay patients were scheduled on each Monday, zero on Tuesday and Wednesday, and three short stay patients were scheduled on each Thursday and Friday. Similarly, one long stay patient was scheduled for surgery on each Tuesday. Moreover, we were able to achieve a 91% occupancy level with only 3.8% cancellations. The results clearly show that a very good scheduling model may lose relevance if it cannot optimise the process for realistic variability in LoS values.

7. Conclusions and future work

In this paper, we have developed a sequential mixed integer programming model to optimise the MSS. It is very difficult to model a real-life process with all of its complexities. However, making unrealistic simplifying assumptions makes a model less useful. In this work, we have made an attempt to model the pa-

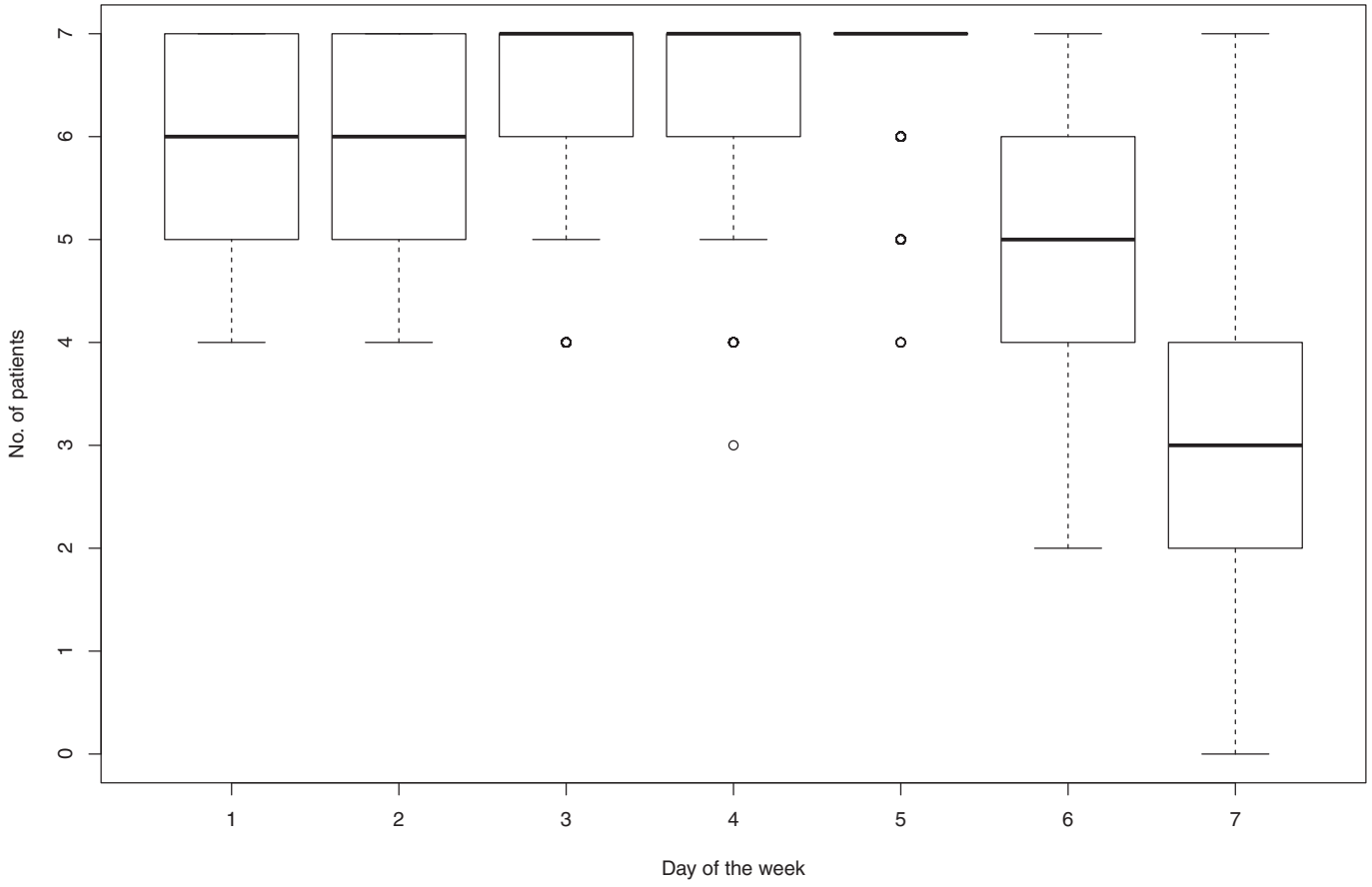


Fig. 3. Boxplot of the number of patients in the ICU on each weekday with MSS_3.

tient flow process without making many assumptions. Our mathematical contribution is a novel approach used in our stochastic MIP model for optimising the MSS in which we used deterministic optimisation on different sequences of randomly-generated LoS scenarios. By conducting various experiments, we demonstrated that our approach is useful for developing a robust MSS without optimising the process over a very large number of LoS scenario realisations. Moreover, we proved that the results (the number of cancellations and the occupancy level) we obtained from the optimisation model were the same as the results obtained by using the simulation model. We also proved that the most frequently optimal MSS was the robust MSS, and not the rounded SAA.

Our model is useful for optimising the MSS of all elective surgery patients. We only presented here the optimal MSS for the patients who need an ICU bed after surgery. However, it can also be used for optimising patient flow in each surgical ward, independently. It can be argued that the optimal solution obtained by optimising each ward independently may not be a feasible solution for a given surgical suite. However, it can provide us with a very good understanding of the process. Moreover, the model can also be extended to optimise the whole surgical suite by increasing the number of patient groups and modelling constraints for each group independently.

The model we have presented in this article is useful for strategic or tactical patient flow management which includes long term decisions. In real life, we also need to make many decisions to manage daily patient flow. For example, we need to cancel overbooked patients' operations according to resource unavailability. We can make these decisions on the day of surgery but that is inconvenient for patients. In future, we are willing to develop a model for operational (day to day) management of patient flow. This model will help us make cancellation decisions before the day of surgery in accordance with the current resource status.

Acknowledgment

We would like to thank Olivia Smith for her ideas to fine tune the solver parameters. Furthermore, this research project is financially supported by an Australian Government Research Training Program Scholarship and Australian Research Council (ARC) linkage grant LP140100152. Peter Taylor would like to acknowledge the support of the ARC through Laureate Fellowship FL130100039 and the ARC Centre of Excellence for the Mathematical and Statistical Frontiers (ACEMS).

Appendix A

A1. Parameters used

g = The number of patient groups.

n = The maximum number of patients allowed to be scheduled in each scheduling period.

m = The number of scheduling periods.

$n_1, n_2, \ldots, n_g$ = The number of patients in each patient group.

$k_1, k_2, \ldots, k_g$ = The maximum number of patients in each group that can be operated on a given day.

beds = The number of beds available in the downstream facility.

Patients: The set of all patients = $\{1, 2, \ldots, n\}$.

$Group_k$: The set of patients in LoS group $k = \{(n_1 + \cdots + n_{k-1}), \ldots, (n_1 + .. + n_k)\}$.

timeblocks: The set of scheduling periods = $\{1, 2, \ldots, m\}$.

days: The set of days in a scheduling period = $\{1, 2, \ldots, 27, 28\}$.

weekdays : The set of weekdays in a scheduling period = {1, 2, 3, 4, 5, 8, 9, 10, 11, 12, 15, 16, 17, 18, 19, 22, 23, 24, 25, 26}.

weekends : The set of weekends or holidays in a scheduling period = {6, 7, 13, 14, 20, 21, 27, 28}.

Available beddays : Total number of beddays available = $28 \times m \times beds$.

$LoS_{t,p}$ = Patient p's LoS in timeblock t.

f: The penalty factor for cancelling a scheduled operations.

PlannedSurgeries = Total number of planned operations.

CancelledSurgeries = Total number of cancelled operations.

Used beddays = Total number of bed days used by patients.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.ejor.2018.04.007.

References

Adan, I., Bekkers, J., Dellaert, N., Vissers, J., & Yu, X. (2009). Patient mix optimisation and stochastic resource requirements: A case study in cardiothoracic surgery planning. *Health Care Management Science, 12*(2), 129–141. doi:10.1007/s10729-008-9080-9.

Beliën, J., & Demeulemeester, E. (2007). Building cyclic master surgery schedules with leveled resulting bed occupancy. *European Journal of Operational Research, 176*(2), 1185–1204. doi:10.1016/j.ejor.2005.06.063.

Blake, J. T., & Carter, M. W. (2002). A goal programming approach to strategic resource allocation in acute care hospitals. *European Journal of Operational Research, 140*(3), 541–561. doi:10.1016/S0377-2217(01)00219-3.

Cappanera, P., Visintin, F., & Banditori, C. (2014). Comparing resource balancing criteria in master surgical scheduling: A combined optimisation-simulation approach. *International Journal of Production Economics, 158*, 179–196. doi:10.1016/j.ijpe.2014.08.002.

Carter, M. W., & Ketabi, S. (2012). Bed balancing in surgical wards via block scheduling. *Journal of Minimally Invasive Surgical Sciences, 1*(4), 129–137. doi:10.5812/jmiss.8084.

Chow, V. S., Puterman, M. L., Salehirad, N., Huang, W., & Atkins, D. (2011). Reducing Surgical Ward Congestion Through Improved Surgical Scheduling and Uncapacitated Simulation. *Production and Operations Management, 20*(3), 418–430. doi:10.1111/j.1937-5956.2011.01226.x.

Cote, M. (2000). Understanding patient flow. *Decision Line*, 8–10.

Figueira, G., & Almada-lobo, B. (2014). Simulation modelling practice and theory hybrid simulation optimization methods: A taxonomy and discussion. *Simulation Modelling Practice and Theory, 46*, 118–134. doi:10.1016/j.simpat.2014.03.007.

Hogg, R. V., Tanis, E. A., & Dale, Z. L. (2015). Chi-square goodness-of-fit tests. In *Probability and statistical inference* (pp. 415–416). New York, NY: Pearson Education, Inc. Ninth edition http://www.mypearsonstore.com/bookstore/probability-and-statistical-inference-9780321923271?xid=PSED.

Kumar, A., & Anjomshoa, H. (2018). A two-stage model to predict surgical patients lengths of stay from an electronic patient database. *IEEE Journal of Biomedical and Health Informatics*. doi:10.1109/JBHI.2018.2819646. in press.

Latouche, G., & Ramaswami, V. (1999). 2. PH distributions. In *Introduction to matrix analytic methods in stochastic modeling: 5* (pp. 33–60). SIAM.

Min, D., & Yih, Y. (2010). Scheduling elective surgery under uncertainty and downstream capacity constraints. *European Journal of Operational Research, 206*(3), 642–652. doi:10.1016/j.ejor.2010.03.014.

Pham, D.-N., & Klinkert, A. (2008). Surgical case scheduling as a generalized job shop scheduling problem. *European Journal of Operational Research, 185*(3), 1011–1025. doi:10.1016/j.ejor.2006.03.059.

Rafaliya, N. R. (2013). *Scheduling elective surgeries in operation room with optimization of post-surgery recovery unit capacity*. The University of Windsor Electronic Theses and Dissertations Paper 4756.

Santos, L. M., Munari, P., Costa, A. M., & Santos, R. H. (2015). A branch-price-and-cut method for the vegetable crop rotation scheduling problem with minimal plot sizes. *European Journal of Operational Research, 245*(2), 581–590. doi:10.1016/j.ejor.2015.03.035.

Tierney, L. T., Hons, B., Conroy, K. M., & Hons, B. (2014). Australian critical care optimal occupancy in the ICU : A literature review. *Australian Critical Care, 27*(2), 77–84. doi:10.1016/j.aucc.2013.11.003.