A dynamic approach to surgical scheduling

Belinda Spratt and Erhan Kozan*

Queensland University of Technology
2 George St, Brisbane, QLD 4000, Australia

Abstract
A mixed integer programming (MIP) formulation is presented that simultaneously tackles operating theatre (OT) planning and scheduling problems through the combined Master Surgical Scheduling (MSS) and Surgical Case Assignment (SCA) problems. We consider stochastic surgical durations and non-elective arrivals whilst applying a dynamic approach to adjust the schedule after cancellations, equipment failure, or new arrivals on the waiting list. The model is based on an Australian public hospital with a large surgical department. Significant detail is included in the formulation to provide practitioners with a model that can be implemented in hospitals. We show that good feasible solutions can be provided in short amounts of computational time with hyper metaheuristics. A dynamic approach is used to show how schedule predictability affects patient throughput. It was found that the use of a two-week schedule increases patient throughput and can help reduce waiting lists.

Keywords: OR in health services; operating theatre planning; robust;

*Corresponding author

e-mail: e.kozan@qut.edu.au
1 Introduction

The efficient utilisation of hospital resources is becoming increasingly important in both private and public hospitals. An increase in patient demand due to aging populations, improved screening techniques and wider access to medical care has led to long waiting times for public patients. Inefficient use of the surgical department can have a major impact on downstream wards.

Operations research techniques have been used to improve the scheduling of surgical procedures with respect to a wide variety of performance measures. For a detailed review on operating theatre (OT) planning and scheduling, including categorization of articles by patient characteristics, performance measures, and decision delineation (among other fields) see Cardoen et al. (2010). Samudra et al. (2016) consider similar fields to those used by Cardoen et al. (2010) in their categorization of OT planning and scheduling literature. Samudra et al. (2016) place particular emphasis on the current challenges and pitfalls observed in recent literature. Guerriero and Guido (2011) review some of the more interesting models and solution techniques used when applying operations research to the planning and scheduling of OTs. OT planning and scheduling problems are typically classified under three main levels: strategic, tactical, and operational. In this paper, we focus on the tactical and operational levels of OT planning and scheduling.

The main problem at the tactical level of OT planning and scheduling is the Master Surgical Scheduling Problem (MSSP). The MSSP is the problem of allocating surgeons, surgical teams, or surgical specialties OT time in the form of a master surgical schedule (MSS). This can be done under an open, block, or modified-block scheduling strategy.

Banditori et al. (2014) present a mixed integer programming (MIP) model for the MSSP. The authors assume a block scheduling policy, and aim to produce a schedule that is not only efficient, but also robust and balanced in terms of resource utilisation.

More recently, Visintin et al. (2016) investigate how the flexibility of surgical teams, operating rooms\(^1\) (ORs) and surgical units can improve the total number of patients scheduled using a MIP formulation of the MSSP. The authors found that allowing surgical teams to change with every new MSS and allowing a mix of both short and long stay cases in the same sessions maximized the number of patients scheduled. Visintin et al. (2016) also found that if flexibility was allowed with respect to both surgical teams and ORs, there was no particular benefit to pooling the resources within surgical units.

\(^1\) An operating theatre (OT) is a set of operating rooms (ORs).
There are two main problems at the operational level: the advanced scheduling and allocation scheduling of the ORs. Advanced scheduling involves determining the list of patients to be seen each day (or time block) in the scheduling horizon. This is also known as the Surgical Case Assignment Problem (SCAP). When solving the SCAP, the scheduling horizon is usually a single week. Bruni et al. (2015) present both a deterministic binary programming model and a stochastic programming model for the SCAP over a one-week scheduling horizon, with an assumed MSS. Their stochastic programming model incorporates non-elective arrivals and random length of stay. The authors present three recourse strategies (overtime, swapping surgeries between ORs, and complete rescheduling) to adjust for disruptions to the schedule.

Molina-Pariente, Hans, et al. (2015) present a number of new heuristics for solving the SCAP. The authors present an ILP model for the SCAP with the objective of minimizing a weighted access time with consideration of patient priority. The authors find that the planning horizon length has a major impact on the performance of the model, and that surgical throughput is highly dependent on the selected objective. Molina-Pariente, Hans, et al. (2015) indicate that further work is required in incorporating stochasticity in the SCAP.

The second problem at the operational level, allocation scheduling, is often performed on a daily basis. Allocation scheduling, or the Surgical Case Sequencing Problem (SCSP), is the sequencing of patients within each day or block. Consideration is often made for children’s surgeries to be earlier in the day or for surgeries with high contamination risk to be performed later in the day. Cardoen et al. (2009) use a weighted normalized objective function to include multiple objectives in the SCSP. The authors consider schedule effects on young, priority, and regional patients.

Although OT planning and scheduling problems are often tackled separately due to computational complexity, solution quality may be improved by a more integrated approach in which problems are solved simultaneously. Testi and Tànfani (2009) provide a binary programming model for the combined MSSP SCAP, taking into account occupancy of downstream units, number of surgical teams available, and patient urgency. Despite the complexity of the problem, the authors are able to use CPLEX to find good solutions to medium sized instances within 600s. The case study considered here is much larger than that considered by Testi and Tànfuni (2009) and includes stochastic surgical durations and non-elective arrivals.

Molina-Pariente, Fernandez-Viagas, et al. (2015) present an MILP for the integrated OT planning and scheduling problem. That is, the problem of assigning a surgical date and OR to a patient (SCAP), and sequencing patients within each OR (SCSP). The authors assume that a team of two surgeons
are present at the surgery, and that surgical duration is dependent on the experience of the assistant surgeon. The authors use simulation to analyse solution robustness.

As each of the OT planning and scheduling problems are NP-hard, metaheuristics and hybrid metaheuristics are often used to produce good feasible solutions in reasonable amounts of computational time. Other approaches include simulation (M’Hallah & Al-Roomi, 2014), and decomposition (Agnetis et al., 2014). Very few papers address a case study as large as that considered herein.

Another trend being seen throughout the literature is the inclusion of stochastic elements in model formulations to produce schedules more robust to uncertainty. Van Riet and Demeulemeester (2015) review existing literature on the trade-offs involved in accounting for non-elective surgeries. The authors state that the main drivers include waiting time, utilisation, overtime, and cancellations. Ferrand et al. (2014) also review the literature on the ways in which the efficiency and responsiveness of the ORs are balanced in the face of non-elective demand.

Stuart and Kozan (2012) developed an innovative reactive scheduling model and solved for a single operating theatre. The model is run in real-time following the completion of each operation and solved with a branch-and-bound heuristic. The criterion was to minimize the weighted number of expected tardy patients.

Saadouli et al. (2015) consider the OT planning and scheduling problem with stochastic surgery and recovery durations. First, the authors use a knapsack formulation to maximize OR utilisation assuming the 85th percentile of surgeries will not exceed the predefined block length. Then, the authors present a MIP model for minimizing the makespan of surgeries.

Heydari and Soudi (2016) propose a two stage stochastic programming formulation with recourse to model the combined SCAP SCSP with non-elective arrivals and uncertain surgical durations. The model is formulated as a hybrid flow shop scheduling problem. The authors assume that hospital administrative staff will be aware of the arrival of any non-elective patients, and their duration, at the start of each day.

Latorre-Núñez et al. (2016) provide a MILP model and constraint programming formulation of the SCAP that simultaneously allocates patients to ORs and plans resource requirements (both staff and beds). The authors use planned slack between elective surgeries to ensure that non-elective patients are treated promptly.

Addis et al. (2016) provide a robust integer linear programming (ILP) formulation of the SCAP under a predefined MSS. The authors use a rolling horizon approach to reschedule on a weekly basis.
whilst taking into account non-elective arrivals, cancellations and new patients on the waiting list. Addis et al. (2016) use a cardinality-constrained approach to allow for variations in surgical durations. The authors perform computational experiments with lognormal realisations of surgical durations on cases with up to 176 patients and either three or four blocks.

Luo et al. (2016) also apply a rolling horizon approach to surgery scheduling. By comparing the performance of a non-rolling horizon model to a rolling horizon model on randomly generated data, the authors find that schedule quality is improved under a rolling horizon model. In particular, the rolling horizon model minimizes OR idle time and increases utilisation. The authors consider only a single specialty in dedicated ORs. Before implementing their rolling horizon model, the authors determine the subset of patients to be operated on throughout the week and schedule days separately by choosing a further subset of potential patients.

Throughout the literature, there is still a need for more realistic models, with fewer simplifications, that can be applied to hospital settings. There is also the need for solution methodologies that can handle long waiting lists and large surgical departments. Currently very few models simultaneously address OT planning and scheduling problems.

To account for stochasticity, we consider lognormally distributed surgical durations and non-elective arrivals. This model maximizes the number of elective surgeries performed, whilst reducing overtime and cancellations. To adapt to the changing waiting list we apply dynamic scheduling approach. In doing so, we produce a robust MSS and SCA over periods of one to four weeks and reschedule every week, limiting the number of deviations from the original schedule. A four week MSS is in line with the case study hospital’s current MSS update frequency. The case study hospital produces SCAs weekly. Thus, considering scheduling horizons of between one and four weeks is compliant with the hospital’s current procedure.

We discuss hyper metaheuristics that can be used to produce good feasible solutions in reasonable amounts of computational time. Importantly, despite the large case study considered, good solutions can be produced using standard desktop computers.

This paper contributes to the literature in a number of ways. Firstly, the model is realistic enough to account for a variety of staff and resource constraints and can be implemented in the hospital environment to improve the current scheduling methodology. The model maximizes surgical throughput whilst adhering to resource limitations. By considering stochastic surgical durations and non-elective arrivals, we are not only able to produce a robust schedule that reduces cancellations and
overtime, but using the dynamic scheduling approach, we are able to reschedule patients in the case of cancellations. In doing so, patient outcomes improve and overall satisfaction increases.

The MIP model presented utilizes the inherent symmetry of the problem to reduce the number of constraints required. The definition of decision variables ensures that the search space is traversed faster than under traditional binary assignment definitions. Innovative hyper metaheuristics are successfully implemented on an OT planning and scheduling model and are shown to produce better solutions than the comparative baseline metaheuristic. Finally, we show that despite the size of the case study, good solutions can be found in short amounts of computational time and the work is therefore accessible to hospital administrators.

In this paper, we address the problem of scheduling patients, surgeons, and specialties in ORs across a multiple weeks. A block scheduling policy is implemented, with either two half-day blocks or one full-day block. We simultaneously consider the MSS and SCA problems.

The problem considered here is based on a case study of a large Australian public hospital. Each year over 100,000 patients are admitted to the hospital, with almost 70,000 admissions to the emergency department (ED). Around 15,000 elective surgical cases are performed each year, along with approximately 6,000 non-elective (emergency and urgent) surgeries. There is a long waiting list for elective surgeries currently composed of almost 2,900 patients.

Elective surgery requests are categorised by urgency and assigned a recommended waiting time of 30, 90, or 360 days. At present, the hospital is able to treat 100% of category one (the most urgent) patients within 30 days of being placed on the waiting list. This falls to around 90% for category two and three patients. We wish to improve the proportion of patients being treated on time, by increasing OT throughput without increasing surgical overtime.

There is capacity for 825 bed spaces at the hospital, around 300 of these being surgical beds. The surgical department has 21 ORs with one reserved for non-elective surgeries, although these tend to overflow into the other ORs. Elective surgeries are typically scheduled on weekdays only and never after eight o’clock at night. Four non-elective ORs are available on weekends. There are around 20 beds in both the surgical care unit (SCU) and post anaesthesia care unit (PACU).

The PAH is at tertiary teaching hospital which results in procedures performed by staff of varying levels of experience. There is limited consultant availability, as many staff members prefer to spend the majority of their time working in private hospitals. The hospital can host visiting medical officers (VMOs) from other hospitals and can reserve OR time for these surgeries. If there is diminished theatre capacity, or an increase in demand, for a particular specialty surgeries can be outsourced to
surrounding hospitals. These intricacies make modelling the OT department quite a complicated process.

The structure of the paper is as follows. The MIP formulation is presented in Section 2.1. We list the assumptions made in Section 2.2. In Section 3, we present the solution techniques including calculation of an upper bound and details of the hyper metaheuristics implemented. Computational results are presented in Section 4. Conclusions and perspectives are discussed in Section 5.

2 The Model
The model presented below is based on a case study of an Australian public hospital with a large surgical department. The model is detailed enough to produce implementable schedules in the hospital under study. The model is also general enough that simple modifications to parameters and constraints will provide a model suitable for use in other hospitals.

We include a wide range of real life constraints including those on surgeon availability and suitability. Surgeon availability data is provided in [dataset] Spratt and Kozan (2017). As this is a teaching hospital, it is important that surgeries are only allocated to surgeons capable of performing the procedure. Surgeons are not allowed to share time blocks. OR suitability is also taken into account when allocating specialties to ORs. For example, expensive equipment should be moved infrequently and it is easier to allocate specialties accordingly.

The model reserves capacity for non-elective patients. This can be adjusted by hospital staff to allow for a variety of risk attitudes. It is assumed that there are no ORs dedicated to non-elective surgeries, but that time blocks can be used by either elective or non-elective patients, but not both. To accommodate the additional non-elective demand, ORs are opened on the weekends for non-elective surgeries only. The number of ORs opened is dependent on surgeon availability, which is based on historical data.

Strum et al. (1998) found surgical durations can be modelled using the lognormal distribution. We assume that both elective and non-elective surgical durations are lognormally distributed. This is verified using transformed historical data using Shapiro-Wilk, Lilliefors, and Anderson-Darling normality tests, and bootstrapping techniques. Due to the availability of data, surgical durations are grouped by specialty. The model does include individual duration parameters for each patient to be used in the case that such data is available. In the case of specialties with highly variable durations, these are split into sub specialties based on patient urgency category.

Patients of the same specialty can share a time block as long as the 95\textsuperscript{th} percentile of their combined surgical durations is at most the length of the block. In the case where a single surgery’s 95\textsuperscript{th} percentile
exceeds the length of a full day block, the surgery can still take place in a full day block, as long as no other surgeries are scheduled for that block. The lognormal parameters used herein are given in [dataset] Spratt and Kozan (2017). Specialties are referred to by number rather than name to respect the confidentiality of hospital data.

Although great care is taken to reduce disruptions by incorporating stochastic surgical durations and non-elective arrivals, we apply a dynamic scheduling approach to reschedule in the case of disruptions. In doing so, we are able to update the schedule to account for cancellations and new arrivals on the waiting list. By producing an initial schedule over a four-week scheduling horizon, we are able to provide some insight into the future workload of hospital staff. The dynamic scheduling approach allows for the flexibility required to create better schedules.

2.1 MIP Model Formulation

In this section, we present an MIP formulation for the combined MSS SCA problem under a dynamic scheduling approach.

Scalar Parameters

\( \bar{H} \): the number of surgeons that practice at the hospital
\( \bar{P} \): the number of patients in the waiting list at the start of the scheduling horizon
\( \bar{S} \): the number of surgical specialties
\( \bar{R} \): the number of ORs
\( \bar{T} \): the number of time periods in the scheduling horizon
\( \bar{W} \): the number of weeks in the scheduling horizon
\( M_{n} \): the maximum number of non-elective patients (of any specialty) that can be seen in a single OR in a full-day block
\( M_{p} \): the maximum number of elective patients (of any specialty) that can be seen in a single OR in a full-day block
\( \xi \): the maximum number of ORs that can be open at any one time during the weekend.

Index Sets

\( H \): the set of surgeons that practice at the hospital. \( H = \{1,\ldots, \bar{H}\} \)
\( P \): the set of patients on the waiting list at the start of the scheduling horizon. \( P = \{1,\ldots, \bar{P}\} \)
\( S \): the set of surgical specialties. \( S = \{1,\ldots, \bar{S}\} \)
\( R \): the set of ORs. \( R = \{1,\ldots, \bar{R}\} \)
\( T \): the set of time periods in the scheduling horizon. \( T = \{1,\ldots, \bar{T}\} \)
$W$ : the set of weeks in the scheduling horizon. $W = \{1, \ldots, \overline{W}\}$

**Indices**

$h$: index for surgeon in set $H$.

$p$: index for patient in set $P$.

$r$: index for OR in set $R$.

$s$: index for surgeon in set $S$.

$t$: index for time period in set $T$.

$w$: index for week in set $W$.

**Vector Parameters**

$E_{ph}$: 1 if patient $p$ can be treated by surgeon $h$, 0 otherwise, $\forall p \in P, h \in H$.

$F_{ht}$: 1 if surgeon $h$ is available during time period $t$, 0 otherwise, $\forall h \in H, t \in T$.

$G_{hs}$: 1 if surgeon $h$ is a member of specialty $s$, 0 otherwise, $\forall h \in H, s \in S$.

$I_{ps}$: 1 if patient $p$ can be treated by specialty $s$, 0 otherwise, $\forall p \in P, s \in S$.

$R_{rs}$: 1 if OR $r$ is equipped for surgeries by specialty $s$, 0 otherwise, $\forall r \in R, s \in S$.

$D_{tt'}$: 1 if time periods $t$ and $t'$ do not overlap, 0 otherwise, $\forall t, t' \in T$.

$B_{t}$: 1 if time period $t$ is a full day block, 0 otherwise, $\forall t \in T$.

$V_{t}$: 1 if time period $t$ is on the weekend, 0 otherwise, $\forall t \in T$.

$U_{tw}$: 1 if time period $t$ is in week $w$, 0 otherwise, $\forall t \in T, w \in W$.

$\kappa^+_s$: the number of elective patients of specialty $s$ that can be treated in a full day block, $\forall s \in S$.

$\kappa^-_s$: the number of elective patients of specialty $s$ that can be treated in a half day block, $\forall s \in S$.

$\hat{\kappa}^+_s$: the number of non-elective patients of specialty $s$ that can be treated in a full day block, $\forall s \in S$.

$\hat{\kappa}^-_s$: the number of elective patients of specialty $s$ that can be treated in a half day block, $\forall s \in S$.

**Decision Variables**

$X_{srt}$: 1 if specialty $s$ or less is assigned to OR $r$, time period $t$, 0 otherwise, $\forall s \in S, r \in R, t \in T$.

$Y_{hrt}$: 1 if surgeon $h$ or less is assigned to OR $r$, time period $t$, 0 otherwise, $\forall h \in H, r \in R, t \in T$.

$Z_{prt}$: 1 if patient $p$ is treated in OR $r$, time period $t$, 0 otherwise, $\forall p \in P, r \in R, t \in T$.

$\Psi_{srt}$: the number of non-elective specialty $s$ patients for OR $r$, time period $t$, $\forall s \in S, r \in R, t \in T$. 
Objective Function

The objective of the dynamic scheduling model is to maximize the number of elective surgeries performed during the scheduling horizon. This will have the effect of reducing the waiting list, decreasing wait time, and improving patient outcomes.

\[
\text{Maximize } \sum_{p \in P} \sum_{r \in R} \sum_{t \in T} Z_{prt} \quad (1)
\]

Constraints

A patient can be treated at most once during the scheduling horizon.

\[
\sum_{r \in R} \sum_{t \in T} Z_{prt} \leq 1, \quad \forall p \in P \quad (2)
\]

If a surgeon is assigned to a time period, then the correct specialty must also be assigned to that time period, in that OR.

\[
Y_{hrt} - Y_{(h-1)rt} \leq \sum_{s \in S} G_{hs} (X_{srt} - X_{(s-1)rt}) , \quad \forall h \in H, r \in R, t \in T \quad (3)
\]

A specialty can only be assigned to an OR if the OR is equipped for that specialty.

\[
X_{srt} - X_{(s-1)rt} \leq R_{rs}, \forall s \in S, r \in R, t \in T \quad (4)
\]

If a patient is assigned to a time period, then the correct surgeon must also be assigned to that time period, in that OR.

\[
Z_{prt} \leq \sum_{h \in H} E_{ph} (Y_{hrt} - Y_{(h-1)rt}), \quad \forall p \in P, r \in R, t \in T \quad (5)
\]

If a patient is assigned to a time period, then the correct specialty must also be assigned to that time period, in that OR.

\[
Z_{prt} \leq \sum_{s \in S} I_{ps} (X_{srt} - X_{(s-1)rt}), \quad \forall p \in P, r \in R, t \in T \quad (6)
\]

If an OR block is reserved for non-elective capacity, then a surgeon must be assigned to that time period. The surgeon assigned must be a member of the appropriate specialty.

\[
\Psi_{srt} \leq M_{s} \sum_{h \in H} (Y_{hrt} - Y_{(h-1)rt}) G_{hs} , \quad \forall s \in S, r \in R, t \in T \quad (7)
\]
A surgeon can only be assigned to a time period if they are available during that time period.

$$\sum_{r \in R} (Y_{hrt} - Y_{(h-1)rt}) \leq F_{ht}, \quad \forall h \in H, t \in T$$

(8)

A single specialty assignment can be made if two time blocks overlap. By enforcing this constraint for $t < \tau$, the symmetry of the constraint is utilized.

$$X_{\text{S}_{rt}} + X_{\text{S}_{\tau r}} \leq 1 + D_{\tau \tau}, \quad \forall r \in R, t, \tau \in T, t < \tau$$

(9)

Based on the 95th percentile of the sum of lognormal surgical durations fitting within a block, we restrict the number of patients assigned to each OR block.

$$\sum_{p \in P} Z_{\text{prt}} I_{ps} \leq (\kappa^+_s B_t + \kappa^-_s (1 - B_t)) (X_{\text{S}_{rt}} - X_{(s-1)rt}), \quad \forall s \in S, r \in R, t \in T$$

(10)

The number of non-elective patients for which capacity is reserved is also limited by the 95th percentile of lognormal surgical durations fitting within the time block.

$$\Psi_{\text{S}_{rt}} \leq (\hat{\kappa}^+_s B_t + \hat{\kappa}^-_s (1 - B_t)) (X_{\text{S}_{rt}} - X_{(s-1)rt}), \quad \forall s \in S, r \in R, t \in T$$

(11)

An OR block can be reserved for either elective patients or non-elective patients, but cannot be shared.

$$\frac{1}{M} \sum_{s \in S} \Psi_{\text{S}_{rt}} \leq 1 - Z_{\text{prt}}, \quad \forall r \in R, t \in T, p \in P$$

(12)

Elective surgery cannot be performed on the weekend.

$$\sum_{p \in P} \sum_{r \in R} Z_{\text{prt}} \leq M_p (1 - V_t), \quad \forall t \in T$$

(13)

Each week, capacity must be reserved for a certain number of non-elective patients of each specialty.

$$\sum_{r \in R} \sum_{t \in T} \Psi_{\text{S}_{rt}} U_{tw} \geq \psi_s, \forall w \in W, s \in S$$

(14)

To ensure that weekends are not relied on too heavily for non-elective surgeries, we limit the number of ORs open at any one time throughout the weekend.

$$\sum_{r \in R} \left( X_{\text{S}_{rt}} + \sum_{\tau \in T} D_{\tau t} X_{\text{S}_{\tau r}} \right) \leq \xi + \bar{R} (1 - V_t), \quad \forall t \in T$$

(15)
2.2 Model Assumptions

In this subsection we discuss the assumptions made when implementing the MSS SCA model on the case study. These assumptions are based on a case study of a large Australian public hospital. It is assumed that historical data is an accurate representation of the number of ORs that can be open, surgeon availability and specialty, surgical duration, and non-elective arrivals.

A modified block scheduling policy is used as per current hospital policy. This means that patients of multiple specialties cannot be treated in the same block. Any ORs that are not reserved under the MSSP can be reallocated or used for non-elective surgeries. No ORs are dedicated to non-elective surgery, but OR blocks may be dedicated to either non-elective or elective surgery. We assume that non-elective patients are prioritized over elective patients. As such, it is assumed that non-elective patients are treated in the first available time block.

Working days are ten hours. We assume that there is no flexibility in the working day, however if overtime is required a surgeon will perform the overtime. If a surgeon is not available for a particular shift, that surgeon will not be scheduled. The surgeon named on the waiting list is the surgeon that must be present during the surgery. If a surgeon is named on the waiting list, then that surgeon may not supervise additional surgeries in different theatres at the time of surgery.

We assume that the suitability of ORs is respected. If a particular specialty does not require specialized equipment, then that surgery is listed as a ‘general’ surgery. We also assume that sufficient capacity exists in downstream wards, ensuring that bottlenecks are highly unlikely.

3 Solution Approaches

The model formulated in Section 2.1 is NP-hard. As such we require the use of metaheuristics to produce solutions in reasonable amounts of computational time. New hyper metaheuristics for use on the OT planning and scheduling problems are developed in this paper. We implement Hyper Simulated Annealing (SA), which selects neighbourhood type through use of a Tabu Search (TS) strategy and accepts solutions using a SA approach. Hyper SA-TS is also implemented, using the Hyper SA algorithm along with a Tabu list of moves. By implementing each of these algorithms, we are able to utilize the problem structure whilst obtaining good solutions in reasonable amounts of computational time. SA is also used in order to compare solution quality and computational effort with the proposed Hyper SA and Hyper SA-TS metaheuristics.

When running the metaheuristics, we consider scheduling horizon lengths of between one and four weeks. Scheduling horizons of between one and four weeks are compliant with the hospital’s current procedure, and can have a large impact on scheduling objectives (Molina-Pariente, Hans, et al., 2015).
The metaheuristic is run, providing a MSS and SCA for the entire scheduling horizon. We assume that the first week goes to plan, and that no patients are cancelled (it is possible to randomly generate cancellations as in Addis et al. (2016)). The patients treated in the first week are removed from the waiting list, and the list is updated with any new arrivals. The schedule calculated in the last iteration is used as an initial schedule for the metaheuristic in the next iteration.

3.1 Baseline Non-elective Schedule

Given the complexity of ensuring sufficient OR capacity is reserved for non-elective surgeries, in this subsection we present a reduced version of the model provided in Section 2.1. This reduced model is used to allocate non-elective OR reservations and is implemented before performing the elective scheduling. This is done in CPLEX. We make a simplifying assumption when creating the baseline non-elective schedule. We assume that non-elective patients will be treated in the first reserved block and that non-elective surgeries are prioritized above elective surgeries.

To ensure that the objective of maximizing the number of elective patients treated is respected; we create the baseline non-elective schedule using the objective shown in (16). This objective is to minimize the amount of OR time reserved for non-elective surgeries that could instead be used for elective surgeries. Since \( X_{S_{rt}} = 1 \) if any specialties were assigned to OR \( r \) during time period \( t \), we minimize (16).

\[
\text{Minimize } \sum_{r \in R} \sum_{t \in T} X_{S_{rt}} \quad (16)
\]

In solving this initial problem, we minimize (16) subject to constraints (2) to (4), (7), (9), (11), (14), and (15).

Fortunately, this problem can be solved by CPLEX in a reasonable amount of time. This solution is used when solving for the elective portion of the schedule using metaheuristics. Although solution quality may be reduced slightly by tackling the non-elective and elective portions of the schedule separately, any reduction in solution quality is negligible compared to the computational effort required to create a full schedule.

3.2 Upper Bound

The upper bound on the number of elective patients that can be treated each week is calculated using CPLEX.

When determining the true upper bound on the total number of patients treated across the six weeks of available data, the waiting lists are combined into a single waiting list.
The upper bound is calculated using the initial non-elective schedule and constraints (2) to (6), (8) to (10), (12), and (13) under the objective shown in (1). IBM ILOG CPLEX Optimization Studio 12.6 was used under a time limit of two hours and the MIP emphasis set to three (improving best bound).

3.3 Hyper SA

The implementation of other metaheuristics has shown that the choice of solution neighbourhood has quite a big impact on solution quality. In order to exploit this fact, we implement a hyper heuristic that uses past neighbourhood performance to determine the neighbourhood to be selected next.

The hyper heuristic uses TS in conjunction with heuristic rankings to select the heuristic to be used for each set of iterations. Solutions are accepted through a SA approach such that improving solutions are always accepted and worsening solutions are accepted according to the change in solution and the solution temperature. In addition to the standard temperature decrease, temperature increases are also performed throughout the run. To account for fewer improving moves being made towards the end of a run, iterations of the same heuristic type are performed in blocks when calculating heuristic rank.

Details of the algorithm are given in the Appendix. The pseudocode of Hyper SA is presented in Algorithm 1, whilst the flow diagram is shown in Figure 1. The Tabu Update subroutine can be seen in Figure 2.

3.4 Hyper SA-TS

After initial investigation showed that the use of TS significantly improved performance in certain problem instances, we include a list of Tabu moves when implementing Hyper SA-TS. A list of the most recent theatre-week-day combinations is stored. The neighbour swap cannot be performed on a Tabu theatre-week-day combination. If a move is made that worsens the solution, the theatre-week-day combination is added to the Tabu list. If the Tabu list exceeds a given length, the oldest Tabu move is removed. Every N iterations the maximum allowable Tabu length is decreased. This gives the metaheuristic more ‘freedom’ in block selection towards the end of the run. It should be noted, that the Tabu theatre-week-day combinations are entirely separate from the list of Tabu neighbourhood types, denoted Tabu_twd and Tabu_n respectively. More sophisticated implementations are possible; however this methodology performs well for this problem. The pseudocode is presented in Algorithm 2, whilst the flow diagram is shown in Figure 3 (see Appendix).

4 Results

In this section, we present the results of computational experiments using SA, Hyper SA, and Hyper SA-TS on a case study of a large Australian public hospital. The surgical department under study is
quite large, with 21 ORs, over 100 surgeons, and 27 specialties (including subspecialties required under the formulation presented in Section 2.1). At present, the hospital has around 2,900 patients on the elective surgery waiting list. The hospital treats approximately 15,000 elective surgery patients per year and a further 6,000 non-elective surgical patients. For specifics on scalar parameters see Table 1.

### Table 1: Initial Scalar Parameter Values

| Parameter | Value |
|-----------|-------|
| \( H \)   | 108   |
| \( \bar{P} \) | 2871  |
| \( \bar{R} \) | 21    |
| \( \bar{S} \) | 27    |
| \( \bar{T} \) | 21    |

Given the size of the surgical department under study, the model presented in Section 2.1 has 180117 binary decision variables and over four million constraints under a one-week scheduling horizon. The size of the problem increases with the scheduling horizon length (c.f. Table 2).

### Table 2: Initial Problem Size

| Scheduling Horizon Length (weeks) | 1     | 2     | 3     | 4     |
|----------------------------------|-------|-------|-------|-------|
| Binary Decision Variables        | 180117| 299943| 419769| 539595|
| Integer Decision Variables       | 11907 | 23814 | 35721 | 47628 |
| Constraints                       | 401233| 7970745| 11938428| 15915372|

The hospital also has a long waiting list for elective surgeries (over 2800 patients). Around 110 non-elective patients are treated at the hospital each week. Details regarding the initial waiting list and average number of surgical requests can be found in [dataset] Spratt and Kozan (2017).

The metaheuristics were coded in MATLAB® and average performance was determined after 100 runs on the university’s HPC facility. It was found that the solution variance had converged sufficiently after 100 runs of each metaheuristic. To demonstrate the ease of implementation in a hospital environment, average computational time was found using MATLAB® on an Intel® Core™ i7-370 CPU @ 3.40GHz with 16 GB of RAM.

The maximum total number of iterations was limited to 16,000 in each case as computational time is a valuable resource in the hospital. Under a one-week scheduling horizon, 16,000 iterations were sufficient for metaheuristic convergence. The other parameters used were found through parameter tuning performed on the university’s HPC facility. In this section, the time shown is the average time required to create the schedules for the entire planning period (scheduling horizon lengths vary) each week for a total of six weeks of data available. We do not include the time required to calculate the
baseline schedule (see Section 3.1) as the baseline schedule is generated once and only ever updated if the predicted non-elective surgery demand changes.

### 4.1 Varying Scheduling Horizon

In this subsection, we consider the effect of varying scheduling horizon length on the mean total number of patients scheduled. Parameter tuning was performed for each of the metaheuristics under each scheduling horizon length. The maximum total iterations were limited to 16,000 in order to keep computational time low whilst ensuring the metaheuristics converge. After thorough parameter tuning, the best performing metaheuristic parameter combinations are given in Table 3.

| Table 3: Metaheuristic Parameters - Varying Scheduling Horizon Length |
|---------------------------------------------------------------------|
| **Metaheuristic** | **Parameter** | **Scheduling Horizon Length (weeks)** |
|                  |               | 1     | 2     | 3     | 4     |
| SA                |               |       |       |       |
| $T$               | 100           | 10    | 10    | 10    |
| $\alpha$          | 0.5           | 0.5   | 0.5   | 0.5   |
| $i_{\text{max}}$  | 50            | 50    | 50    | 50    |
| Clean             | M             | M     | M     | M     |
| Hyper SA          |               |       |       |       |
| $T$               | 1             | 1     | 1     | 1     |
| $\beta$           | 0.95          | 0.9   | 0.85  | 0.95  |
| $i_{\text{max}}$  | 500           | 500   | 500   | 500   |
| Tabu*             | 4             | 4     | 2     | 2     |
| Hyper SA-TS       |               |       |       |       |
| $T$               | 1             | 100   | 100   | 10    |
| $\beta$           | 0.95          | 0.85  | 0.85  | 0.85  |
| $i_{\text{max}}$  | 500           | 500   | 500   | 500   |
| Tabu*             | 3             | 3     | 4     | 4     |

Results from these computational experiments are displayed in Table 4. Table 4 includes the mean total number of patients scheduled, the variance, worst and best objectives, and the time taken (in seconds) to schedule all six weeks of historical data, whilst implementing a dynamic scheduling approach with scheduling horizon lengths between one and four weeks. For example, under a four week scheduling horizon we produce a four week schedule, update waiting lists assuming the first week went to plan, and reschedule for another four week scheduling horizon until all six weeks of historical data have been scheduled.
The results displayed in Table 4 indicate that Hyper SA is the best performing metaheuristic under scheduling horizons of one to four weeks. Hyper SA appears to outperform Hyper SA-TS under scheduling horizons of two ($p<0.01$), three ($p<0.01$), and four ($p<0.05$) weeks.

The upper bound is calculated assuming all waiting lists are known in advance. The upper bound did not converge within 7200 seconds. The upper bound was 1451.17, whilst the best feasible integer solution found during this time was the trivial empty schedule.

The actual number of elective patients treated across the six weeks was 1228, thus using the methodology presented in this paper we are able to clear a significant amount of backlog. We schedule approximately 24 additional patients each week whilst reserving more than enough capacity for non-elective patients. This is done while also ensuring number of ORs in use is in-line with historical averages. If the hospital were to consider a higher risk attitude toward overtime, even more patients could be scheduled.

Two-sample t-tests indicate that Hyper SA ($p<0.01$) and Hyper SA-TS ($p<0.05$) performs best under a two-week scheduling horizon. SA performs best under a scheduling horizon of one week ($p<<0.01$). This may be due to the increase in solution space associated with a longer scheduling horizon, without any increase in maximum total iterations. To further investigate this, parameter tuning is performed for each of the metaheuristics with maximum total iterations now 16,000 times the length of the scheduling horizon (in weeks). The top performing metaheuristic parameter combinations are shown in Table 5. The new computational results are displayed in Table 6.
Table 5: Metaheuristic Parameters - Varying Maximum Iterations

| Metaheuristic   | Parameter | Scheduling Horizon Length (weeks) |
|-----------------|-----------|----------------------------------|
|                 |           | 1      | 2      | 3      | 4      |
| SA              | $T$       | 100    | 10     | 10     | 10     |
|                 | $\alpha$  | 0.5    | 0.5    | 0.5    | 0.5    |
|                 | $i_{max}$ | 50     | 50     | 50     | 50     |
|                 | Clean     | M      | M      | M      | M      |
| Hyper SA        | $T$       | 1      | 10     | 1      | 10     |
|                 | $\beta$   | 0.95   | 0.95   | 0.85   | 0.85   |
|                 | $i_{max}$ | 500    | 500    | 500    | 500    |
|                 | Tabu*     | 4      | 3      | 3      | 4      |
| Hyper SA-TS     | $T$       | 1      | 1      | 1      | 1      |
|                 | $\beta$   | 0.95   | 0.95   | 0.9    | 0.95   |
|                 | $i_{max}$ | 500    | 500    | 500    | 500    |
|                 | Tabu*     | 3      | 3      | 4      | 2      |

When the total number of iterations is increased proportionally to scheduling horizon length, Hyper SA remains the top performing metaheuristic under scheduling horizons of one, two, and three weeks, whilst Hyper SA-TS is the top performing metaheuristic using a four-week scheduling horizon approach. Under scheduling horizons of three and four weeks, two-sample t-tests indicate that there may be no significant difference between Hyper SA and Hyper SA-TS ($p > 0.05$).

Table 6 shows that the metaheuristics display a significant increase in number of patients scheduled when the number of iterations is increased proportionally to scheduling horizon length. Overlapping 95% confidence intervals and two-sample t-tests ($p > 0.05$) show that under the increased total iterations Hyper SA and Hyper SA-TS may not perform significantly better under a scheduling horizon of three weeks compared to a two-week scheduling horizon. Under the increased total iterations, SA performs significantly better with a scheduling horizon of two weeks ($p < 0.01$).
**Table 6: Computational Results – Effect of Increasing Total Iterations**

| Horizon Length (weeks) | Method       | Mean     | Variance | Worst  | Best  | Time (seconds) |
|------------------------|--------------|----------|----------|--------|-------|----------------|
| 1                      | SA           | 1359.34  | 25.78    | 1346   | 1371  | 96.53          |
|                        | Hyper SA     | 1361.70  | 27.02    | 1348   | 1372  | 103.49         |
|                        | Hyper SA-TS  | 1361.63  | 23.57    | 1349   | 1373  | 113.29         |
| 2                      | SA           | 1365.45  | 26.51    | 1351   | 1376  | 190.50         |
|                        | Hyper SA     | 1369.34  | 16.69    | 1359   | 1379  | 232.17         |
|                        | Hyper SA-TS  | 1368.23  | 16.91    | 1360   | 1378  | 243.58         |
| 3                      | SA           | 1363.35  | 24.92    | 1344   | 1375  | 298.75         |
|                        | Hyper SA     | 1369.55  | 14.90    | 1360   | 1379  | 363.24         |
|                        | Hyper SA-TS  | 1368.87  | 13.29    | 1361   | 1379  | 393.82         |
| 4                      | SA           | 1347.03  | 37.36    | 1323   | 1358  | 411.41         |
|                        | Hyper SA     | 1359.22  | 22.92    | 1345   | 1369  | 487.58         |
|                        | Hyper SA-TS  | 1359.39  | 19.82    | 1346   | 1367  | 515.55         |

It can be seen that the increased throughput is due to an increase in OT utilisation and the choice of surgical specialty. Preference is given to surgical specialties with shorter surgeries. This is in line with the shortest first heuristic rules which tend to perform well on scheduling problems. For example, ophthalmology is allocated many time blocks, as they are able to treat up to eight patients in a full-day block. After this initial clearing of backlog, lower weekly throughput is expected as bottlenecks become more apparent.

There is a significant reduction in overtime by considering the 95\textsuperscript{th} percentile of surgical durations. Using the 95\textsuperscript{th} percentile of lognormal surgical durations, we do not see any overtime caused by elective surgeries. It is expected that overtime may occur once throughout the week during the time reserved for liver transplants. This is as the 95\textsuperscript{th} percentile of the duration of a single liver transplant is approximately 11.82 hours. Thus, the total overtime seen in the new schedule is only 1.82 hours. This is only 0.19\% of the 950.54 hours scheduled.

In comparison, a total of 54.22 hours of surgery was performed outside of 8am to 6pm in the historical schedule. This is approximately 7.31\% of the 741.80 surgical hours used throughout the week. This difference in overtime may be due to the difference in how capacity is reserved for non-elective surgeries. At present, the hospital dedicates OR8 and OR10 to non-elective surgeries. The majority of overtime occurs in these ORs. In the model presented in this paper, we do not reserve specific ORs for non-elective surgeries and instead reserve a variety of ORs throughout the week. In reality, surgeries may be too urgent to hold until the next morning and will instead be performed as soon as possible. This would increase the overtime observed when implementing the new schedule.
5 Conclusion
In this paper we presented a MIP formulation of the combined MSSP SCAP. The model includes stochastic surgical durations, non-elective arrivals and a dynamic scheduling approach. In this model, we were able to address the presence of non-elective patients by reserving OT capacity in blocks. In doing so, we also allowed non-elective surgeries to be performed on weekends. The surgical durations of both elective and non-elective patients are well modelled by the lognormal distribution. As such, constraints ensured that the 95th percentile of surgical durations does not exceed the length of the time block. In allocating time for non-elective surgeries, we also ensure that these time blocks are not shared by elective patients.

We consider a number of solution techniques including metaheuristics and hyper metaheuristics. In particular we find that Hyper SA and Hyper SA-TS perform very well compared to the upper bound, and require significantly less time. The upper bound solution also required significant amounts of pre and post processing to produce sensible solutions.

One of the model limitations lies in the way in which non-elective capacity is allocated. It is assumed that hospital administrators will shift elective patients to non-elective blocks if highly urgent cases arrive, allowing non-elective surgeries to be performed in the newly empty elective block. To address this limitation, future work includes producing and validating a number of simple heuristics for the reactive rescheduling of the ORs in the case of schedule disturbances.

Acknowledgements
This research was funded by the Australian Research Council (ARC) Linkage Grant LP 140100394. Computational resources and services used in this work were provided by the HPC and Research Support Group, Queensland University of Technology, Brisbane, Australia.

References

Addis, B., Carello, G., Grosso, A. & Tànfani, E. (2016). Operating room scheduling and rescheduling: a rolling horizon approach. Flexible Services and Manufacturing Journal, 28(1-2), 206-232.

Agnetis, A., Coppi, A., Corsini, M., Dellino, G., Meloni, C. & Pranzo, M. (2014). A decomposition approach for the combined master surgical schedule and surgical case assignment problems. Health Care Management Science, 17(1), 49-59.

Banditori, C., Cappanera, P. & Visintin, F. (2014). Investigating the Relationship Between Resources Balancing and Robustness in Master Surgical Scheduling. In: A. Matta, J. Li, E. Sahin, E. Lanzarone & J. Fowler, Proceedings of the International Conference on Health Care Systems Engineering (Vol. 61, pp. 149-162): Springer International Publishing.
Bruni, M.E., Beraldi, P. & Conforti, D. (2015). A stochastic programming approach for operating theatre scheduling under uncertainty. IMA Journal of Management Mathematics, 26 (1), 99-119.

Cardoen, B., Demeulemeester, E. & Beliën, J. (2009). Optimizing a multiple objective surgical case sequencing problem. International Journal of Production Economics, 119 (2), 354-366.

Cardoen, B., Demeulemeester, E. & Beliën, J. (2010). Operating room planning and scheduling: A literature review. European Journal of Operational Research, 201 (3), 921-932.

Ferrand, Y.B., Magazine, M.J. & Rao, U.S. (2014). Managing operating room efficiency and responsiveness for emergency and elective surgeries - A literature survey. IIE Transactions on Healthcare Systems Engineering, 4 (1), 49-64.

Guerriero, F. & Guido, R. (2011). Operational research in the management of the operating theatre: a survey. Health Care Management Science, 14(1), 89-114.

Heydari, M. & Soudi, A. (2016). Predictive / Reactive Planning and Scheduling of a Surgical Suite with Emergency Patient Arrival. Journal of Medical Systems, 40(30), 1-9.

Latorre-Núñez, G., Lüer-Villagra, A., Marianov, V., Obreque, C., Ramis, F. & Neriz, L. (2016). Scheduling operating rooms with consideration of all resources, post anesthesia beds and emergency surgeries. Computers & Industrial Engineering, 97, 248-257.

Luo, L., Luo, Y., You, Y., Cheng, Y., Shi, Y. & Gong, R. (2016). A MIP Model for Rolling Horizon Surgery Scheduling. Journal of Medical Systems, 40(5), 1-7.

M’Hallah, R. & Al-Roomi, A.H. (2014). The planning and scheduling of operating rooms: A simulation approach. Computers & Industrial Engineering, 78, 235-248.

Molina-Pariente, J.M., Fernandez-Vigias, V. & Framinan, J.M. (2015). Integrated operating room planning and scheduling problem with assistant surgeon dependent surgery durations. Computers & Industrial Engineering, 82, 8-20.

Molina-Pariente, J.M., Hans, E.W., Framinan, J.M. & Gomez-Cia, T. (2015). New heuristics for planning operating rooms. Computers & Industrial Engineering, 90, 429-443.

Saadouli, H., Jerbi, B., Dammak, A., Masmoudi, L. & Bouaziz, A. (2015). A stochastic optimization and simulation approach for scheduling operating rooms and recovery beds in an orthopedic surgery department. Computers & Industrial Engineering, 80, 72-79.

Samudra, M., Van Riet, C., Demeulemeester, E., Cardoen, B., Vansteenikste, N. & Rademakers, F.E. (2016). Scheduling operating rooms: achievements, challenges and pitfalls. Journal of Scheduling, 19 (5).

[dataset] Spratt, B. & Kozan, E. (2017). Australian public hospital surgical case study. In: (Vol. 1): Mendeley Data.

Strum, D.P., May, J.H. & Vargas, L.G. (1998). Surgical procedure times are well modeled by the lognormal distribution. Anesthesia & Analgesia, 86(2S), 47S.
Stuart, K. & Kozan, E. (2012). Reactive scheduling model for the operating theatre. Flexible Services and Manufacturing Journal, 24 (4), 400-421.

Testi, A. & Tànfani, E. (2009). Tactical and operational decisions for operating room planning: Efficiency and welfare implications. Health Care Management Science, 12 (4), 363-373.

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.

Visintin, F., Cappanera, P. & Banditori, C. (2016). Evaluating the impact of flexible practices on the master surgical scheduling process: an empirical analysis. Flexible Services and Manufacturing Journal, 28(1-2), 182-205.
Appendix

**Algorithm 1: Hyper SA**

Set cost\_p = 0, i = 0 and nrep = 1.

**WHILE** i <= Maximum Iterations
  Set neighbourhood to the non-Tabu neighbourhood with the highest rank.
  Set cost\_c = 0.
  **FOR** j = 1 **TO** nrep
    i = i + 1;
    Find a random neighbour.
    Calculate objective.
    **IF** objective > cost\_c
      Set cost\_c = objective.
    **END**
    Check feasibility.
    **IF** objective > best objective and solution is feasible
      Update best solution and objective.
    **END**
    Calculate acceptance probability function.
    **IF** acceptance probability > rand and solution is feasible
      Update solution.
      Reduce temperature.
    **ELSE**
      Increase temperature.
    **END**
  **END**
  **IF** mod(i, max repetitions per temperature ) == 0
    Reduce r by a factor of gamma\_star.
    Recalculate alpha.
    **IF** i == max repetitions per temperature
      Set nrep = max repetitions per temperature.
    **END**
  **END**
  **IF** cost\_c > cost\_p
    Increase the rank of the current neighbourhood type by 1.
  **ELSE**
    Decrease the rank of the current neighbourhood type by 1.
    Set the current neighbourhood type to Tabu.
    If the Tabu List is too long, remove the oldest neighbourhood from the list.
  **END**
  Set cost\_p = cost\_c.
**END**
Figure 1: Hyper SA flow diagram.

Figure 2: Tabu Update flow diagram.
Algorithm 2: Hyper SA-TS

WHILE i <= Maximum Iterations
    Set neighbourhood to the non-Tabu_n neighbourhood with the highest rank.
    Set cost_c = 0.
    FOR j = 1 TO nrep
        i = i + 1;
        Select a non-Tabu_twd theatre-week-day combination.
        Find a random neighbour. Calculate objective.
        IF solution not improved
            Add theatre-week-day combination to Tabu_twd.
            IF Tabu list is too long
                Remove oldest theatre-week-day combination from Tabu_twd.
        END
    END
    IF objective > cost_c
        Set cost_c = objective.
    END
    Check feasibility.
    IF objective > best objective and solution is feasible
        Update best solution and objective.
    END
    Calculate acceptance probability function.
    IF acceptance probability > rand and solution is feasible
        Update solution. Reduce temperature.
    ELSE
        Increase temperature.
    END
    IF mod(i, max repetitions per temperature) == 0
        Reduce r by a factor of gamma_star.
        Recalculate alpha.
        IF i == max repetitions per temperature
            Set nrep = max repetitions per temperature.
    END
    IF cost_c > cost_p
        Increase the rank of the current neighbourhood type by 1.
    ELSE
        Decrease the rank of the current neighbourhood type by 1.
        Set the current neighbourhood type to Tabu_n.
        If Tabu_n is too long, remove the oldest entry.
    END
    Set cost_p = cost_c.
    IF mod(iter_num, N) == 0
        Reduce the length of the Tabu_twd List.
    END
END
Figure 3: Hyper SA-TS Flow Diagram.