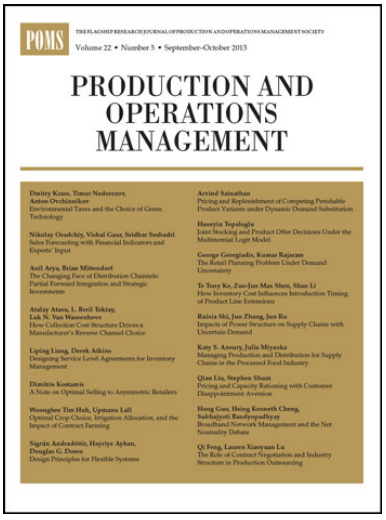
Accepted Manuscript



Title: Increased surgical capacity without additional resources: Generalized operating room planning and scheduling

Author: Bahman Naderi, Vahid Roshanaei, Mehmet A. Begen, Dionne M. Aleman, David R. Urbach

DOI: https://doi.org/doi:10.1111/poms.13397
Reference: POMS 13397
To appear in: Production and Operations Management

Please cite this article as: Naderi Bahman.,et.al., Increased surgical capacity without additional resources: Generalized operating room planning and scheduling. *Production and Operations Management* (2021), https://doi.org/doi:10.1111/poms.13397

This article has been accepted for publication and undergone full peer review but has not been through the copyediting, typesetting, pagination and proofreading process, which may lead to differences between this version and the Version of Record. Please cite this article as doi: 10.1111/poms.13397

Increased surgical capacity without additional resources: Generalized operating room planning and scheduling

Bahman Naderi[1], Vahid Roshanaei[2], Mehmet A. Begen[3], Dionne M. Aleman[4], and David R. Urbach[5]

[1]Mechanical, Automotive, and Materials Engineering, Faculty of Engineering, University of Windsor, Windsor, Canada
[2]Department of Operations Management & Statistics, Rotman School of Management, University of Toronto, Canada
[3]Corresponding author: mbegen@ivey.uwo.ca - Ivey Business School, Western University, London, Canada
[4]Department of Mechanical and Industrial Engineering, University of Toronto, Canada
[5]Toronto General Hospital, General Surgery Department, Toronto, Canada

Abstract

We study a generalized operating room planning and scheduling (GORPS) problem at the Toronto General Hospital (TGH) in Ontario, Canada. GORPS allocates elective patients and resources (i.e., operating rooms, surgeons, anesthetists) to days, assigns resources to patients, and sequences patients in each day. We consider patients' due-date, resource eligibility, heterogeneous performances of resources, downstream unit requirements, and lag times between resources. The goal is to create a weekly surgery schedule that minimizes fixed- and over-time costs. We model GORPS using mixed-integer and constraint programming models. To efficiently and effectively solve these models, we develop new' multi-featured logic-based Benders decomposition approaches. Using data from TGH, we demonstrate that our best algorithm solves GORPS with an average optimality gap of 2.71% which allows us to provide our practical recommendations. **First**, we can increase daily OR utilization to reach 80%—25% higher than the status quo in TGH. **Second**, we do not require to optimize for the daily selection of anesthetists —this finding allows for the development of effective dominance rules that significantly mitigate intractability. **Third**, solving GORPS without downstream capacities (like many papers in literature) makes GORPS easier to solve, but such OR schedules are only feasible in 24% of instances. **Finally**, with existing ORs' safety capacities, TGH can manage 40% increase in its surgical volumes. We provide recommendations on how TGH must adjust its downstream capacities for varying levels of surgical volume increases (e.g., current urgent need for more capacity due to the current Covid-19 pandemic).

Keywords: Healthcare operations, planning and scheduling, operating rooms, multiple resources, performance heterogeneity, overtime, downstream capacities, logic-based Benders decomposition.

History: Received: August 2020; accepted February 2021 by Sergei Savin after a revision.

1 Introduction

Operating rooms (ORs) play a substantial role in hospital profitability, and their optimal utilization is vital in containing the cost of surgical service delivery, shortening surgical patient wait times, and increasing patient admissions (Roshanaei et al., 2017b). Since many OR resources are usually considered fixed expenses, improving throughput just by one additional procedure per day per OR can generate anywhere from $4-7 million in additional annual revenue for an average-sized organization (HFMA, 2005). We study the problem of generalized OR planning and scheduling (GORPS), which we define as elective surgery scheduling

where a pool of elective surgeries within a single specialty, must be operated on during a finite planning horizon using existing resources.

Each patient joins the surgical wait list after the decision to have surgery (decision-to-treat) is made. At this point, we know the following details for each patient on the wait list: the type of required surgery, the set of eligible surgeons and ORs, ORs downstream requirements, and a maximum wait time (due-date). GORPS encompasses decisions regarding high-level planning of resources and detailed scheduling of surgeries. More specifically, GORPS optimizes four decisions (i) *allocation* of patients to days, (ii) *allocation* of members of resources to days, (iii) *assignment* of members of resources to patients for each day, and (iv) *sequencing/scheduling* of surgeries assigned to each member of each resource in each day. The first two decisions occur in the *planning phase* and the next two occur in the *scheduling phase*. The objective function is to minimize the total fixed- and over-time costs associated with using different members of different resources. We consider three resources: ORs, surgeons, and anesthetists—but the approach is generalizable to any number of resources. Each resource can consist of a different number of members. We can treat overtime of each resource as a continuous or step function.

Each member of a resource has a specified set of working days in that s/he works, which is at least eight regular hours and includes up to two hours of overtime per day, if need arises. Each member of a resource works exclusively for the same surgical specialty and resources are not shared among different surgical specialties. Surgeons and anesthesiologists can be assigned to more than one OR in each day. We therefore use an *open scheduling strategy* to manage the availability times of these two resources and ensure that no two surgeries have the same starting times for the same surgeon and anesthesiologist. The schedule that we create must enforce a lag time (minimum, maximum, or no time lag) between the starting and finishing times of different members of different resources for a patient. For example, an anesthesiologist can start inducing the anesthesia only after nurses have prepared the patient for operation. A surgeon can commence a surgery only after anesthesia has been administered by an anesthesiologist. The goal of our optimization is to design a realistic schedule that adheres to the resource availability/eligibility, heterogeneous performances of different members of different resources, and pre-determined lag times among resources while minimizing the fixed and variable operating costs of resources. The optimization model that we develop for GORPS considers parameters, including surgical durations, as deterministic because computational intractability of GORPS makes impossible the consideration of even small number of stochastic scenarios.

The techniques and insights of our paper are of paramount importance for healthcare practitioners since they are related to several ongoing and recent pressing issues, namely demand for scarce healthcare resources (such as ICU beds) is increasing faster than capacity improvements of these resources. There is a recent report that alludes to the naturally growing population of seniors in Toronto (CBCNews, 2017): Seniors constitute 15% of the city's population and in 2016 this number increased by 33.5% from 1996 and by 13.1% from 2011. Despite such an increase in elderly population in Toronto, capacity improvements in healthcare recourses that these patients need, especially supply of ICU beds has significantly lagged behind, leading to longer wait time for all surgical patients. The ongoing COVID-19 pandemic has severely exacerbated the need for ICU beds, calling on further allocation of ICU beds to COVID-19 patients, rendering ICU beds as invaluable life-saving resources. Grappled with the enormity of health implications of COVID-19, almost all healthcare institutions in Canada (and similarly around the world) have decided to prudently allocate majority of their ICU beds to COVID-19 patients and this issue has culminated in cancellation of nearly 400,000 elective surgeries across Canada by mid-June 2020 (GlobalNews, 2020). Canadian provinces have begun addressing the unprecedented backlog of elective surgeries in wake of COVID-19 (CTVNews, 2020):

"It could take well over a year to clear [the backlog of elective surgeries]". The situation is not different for other countries as a recent paper (COVIDSurg-Collaborative, 2020) shows that there were over 25 million surgeries canceled world-wide during the peak 12 weeks of COVID-19 pandemic and it would take about 45 months to clear the backlog if countries increase their surgeries by 20%. A more shocking news is the association between brain and cardiac strokes in youth (without any previous symptoms) due to COVID-19 (Oxley and Mocco J., 2020). Stroke patients require prompt treatment in ICU beds of hospitals. Allocating ICU beds to such high-priority patients further constrains the admission of elective surgical patients whose ICU hospitalization is part of their surgery pathway. In view of such demand increase for ICU beds, hospitals must deploy viable strategies to readmit and accommodate these elective surgical patients as quickly as possible. We will discuss trade-offs that hospitals must consider to ensure higher cost-effectiveness and patient outcome later in our paper.

We summarize our contributions to the literature as follows:

1. **Problem definition**. We study an operating room planning and scheduling for the General Surgery Department at the Toronto General Hospital in Ontario, Canada. Models and methods that we develop for this problem might be considered as a generalization of an existing integrated OR-anesthetist scheduling problem (Rath et al., 2017) in that we encompass the allocation, assignment, and sequencing of *any number of resources* in an OR while capturing their heterogeneous performances and lag times between their starting and finishing times on any surgery. Another generalization is to model flexibility of anesthesiologists' availability for surgeries that allows the decision maker to optimally determine for which portion of a surgery an anesthesiologist must be present. While we are legally obliged to follow Ontario guidelines for determining an anesthesiologist's availability for each surgery, we present a flexible framework that allows OR managers to capture availability however they deem fit according to the governing rules and regulations in their hospitals. We additionally consider the downstream bed requirements for each surgical procedure, including length of stays in post-anesthesia care unit (PACU), intensive care unit (ICU), and ward beds. Unlike existing models in the literature that model overtime as a continuous function (Denton et al., 2010; Rath et al., 2017; Roshanaei et al., 2017b), we treat daily overtime of each member of each resource as either step function or continuous function, because in some hospitals overtime is allocated in a specified block of time. For example, if the block is 15 minutes, it means that one minute of overtime equals 15 minutes and 16 minutes of overtime equals 30 minutes. Thus, compared to related papers in the literature, GORPS captures more realistic features that are useful in hospitals.

2. **Models and methods**. To solve GORPS, we first develop mixed-integer programs (MIPs) using one of the four popular sequence modeling paradigms in the general scheduling literature: (i) position-based (PB) (Wagner, 1959), (ii) sequence-based (SB) (Manne, 1960), (iii) immediate-sequence-based (ISB) (Wilson, 1989), and (iv) time-based (TB) (Bowman, 1959). In addition to MIPs, we formulate GORPS as a novel constraint programming (CP) model. We extend the MIP and CP models to treat the overtime of each member of each resource as a step function rather than as a continuous function. For OR scheduling, we conduct the first comprehensive evaluation among the performances of these MIP and CP models and investigate *trade-offs between decision-making speed and solution quality* of MIPs and CP. We demonstrate that some of the MIP models cannot even find a feasible solution and those able to solve the problem produce large optimality gaps within a reasonable runtime.

   To enhance solvability of GORPS, we develop several logic-based Benders decomposition methods (LBBDs) that encompass entirely unique decomposition schemes, partitioning GORPS into smaller

and more manageable components. Our LBBDs encompass novel partitioning strategies that efficiently manage computational difficulty associated with performance heterogeneity of resources. These LBBDs partition GORPS into integrated allocation(-assignment) master problem (MP), multiple sequencing/scheduling sub-problems (SPs) (one for each day) and connect them via novel Benders feasibility and optimality cuts. We design novel non-linear SP relaxations to enhance the quality of integer solutions that MPs in different LBBDs find. Since the inclusion of these non-linear relaxations gives rise to *mixed-integer nonlinear programs*, we develop a novel linearization scheme that uses auxiliary *continuous* variables, significantly enhancing solution quality without adding substantial computational burden. To solve the SPs of these LBBDs, we develop MIP and CP models and discuss which one works the best. We show that LBBDs can be decomposed into two novel ways: (i) *variable-based* (most literature uses a similar concept), and (ii) *resource-based*, which constitutes a novel feature of this study. We compare the decomposition schemes and show their convergence rates as a function of Benders cuts, which are of varying strengths. We also implement the cutting-plane variants of these LBBDs, called Branch-and-Check (B&C). We believe heterogeneity-based exact decomposition design is a novel contribution of our work that can be used to solve similar optimization problems, not only in healthcare but also in other industries such as transportation, supply chain, and manufacturing in which resources are mostly heterogeneous.

3. **Data analysis and managerial insights**. We conduct a descriptive and diagnostic analyses on our data, which has been collected between June 2011 to June 2013 from the General Surgery Department of TGH. We have records of 2711 elective patients. We analyze distribution for surgical durations, length of stays of patients in downstream units, typical routes that patients take through the operating theatre, daily and weekly average number of elective surgeries, and average utilization of surgeons and ORs. We also capture heterogeneity in performances of resources (surgeons in particular) for different types of surgeries. We incorporate these findings as features into our optimization models. Using various instances of this dataset, we ascertain which of our algorithms is best suited for designing TGH's weekly surgical schedule. Having ascertained the best algorithm, we use it to provide our practical recommendations. First, we show that we can increase utilization of all resources and allow daily OR utilization to reach 80%—25% higher than the status quo in TGH. Second, we counter-intuitively demonstrate that we do not require to optimize for the daily number and selection of anesthetists (using binary variables); we instead only require to determine their daily overtime (using continuous variables)—this finding allowed us to develop effective dominance rules that significantly mitigated intractability and simplified analysis. Third, we empirically show that solving GORPS without downstream capacities (like many papers in the literature) makes GORPS easier to solve, but such constructed schedules are feasible in only 24% of instances. Finally, we demonstrate that with existing ORs' safety capacities, TGH can manage 40% increase (10 more patients per week, on average, e.g., due to aging population and readmission of cancelled elective surgeries due to COVID-19) in its surgical volumes if downstream capacities are ignored. We provide recommendations on how TGH must adjust its downstream capacities for varying levels of surgical volume increases and discuss the trade-offs thereof. We believe these insights obtained for TGH can be used for other similar hospitals.

We present our data analyses, some of our models and their parametric size complexities, and proofs in the Appendix.

2 Literature review

We divide the literature review into two parts: mathematical modeling paradigms for sequencing problems and solution methods for OR scheduling problems.

2.1 Mathematical modeling paradigms for sequencing problems

Operating room managers in hospitals require an effective scheduling tool that provides quality solutions within a reasonable computation time. Mathematical models are able to achieve a good trade-off between these two factors if the size of the problem is not too large (Santibanez et al., 2007). Mixed-integer scheduling model development for manufacturing operations is an active field of research (see Ku and Beck (2016); Naderi and Ruiz (2010); Pan (1997); Roshanaei et al. (2013); Stafford et al. (2004) for popular sequence modeling paradigms). There are four mainstream mathematical modeling paradigms for formulating a sequencing problem: (i) sequence-based (SB) (Manne, 1960), (ii) immediate-sequence-based (ISB), (iii) position-based (PB) (Wagner, 1959), and (iv) time-based (TB) (Bowman, 1959). There is no consensus as to which one of these mathematical models performs better than others. The computational performance unpredictability in sequencing paradigms has spurred many studies to compare each sequencing modeling paradigm for popular manufacturing operations settings, including flow-shop (Pan, 1997; Stafford et al., 2004), distributed flow shop (Naderi and Ruiz, 2010), job shop (Ku and Beck, 2016; Pan, 1997), open shop (Naderi et al., 2011), and flexible job shop (Roshanaei et al., 2013) scheduling problems. Most of these studies are focused on the first three sequence modeling paradigms because the performance of time-based models varies depending on how surgical (processing) and resource availability times are discretized. Different choices of time discretization can result in high variability in the size of the underlying sequencing models, which are mostly formulated as integer programs rather than mixed-integer programs; the optimal solution of the time-based sequencing models will be an approximation of the problem (Castro and Marques, 2015). For the sake of completeness and future use by other researchers, we also develop time-based sequencing models for GORPS that uses a minute-based (same across all models) time discretization.

Surgery sequence optimization within an OR has already been studied using sequence-based (Batun et al., 2011; Roshanaei et al., 2017b), immediate-sequence-based (Hashemi Doulabi et al., 2020), time-based (Hashemi Doulabi et al., 2016), and constraint programming (Wang et al., 2015) modeling paradigms. Constraint programming (CP) is another paradigm for modeling surgery sequencing, and it has been shown to be relatively effective for solving integrated allocation-sequencing OR scheduling problems for realistically-sized instances of the problem (Hashemi Doulabi et al., 2016). We show, however, that CP can be incorporated effectively into our decomposition techniques to quickly ascertain surgical schedule feasibility. To the best of our knowledge, there is no systematic comparative evaluation among all of these modeling paradigms, especially on large-scale optimization problem like GORPS.

Denton et al. (2010) developed deterministic, stochastic, and robust MIPs for a single-resource OR scheduling problem to minimize fixed and overtime OR costs. Rath et al. (2017) extended the work of Denton et al. (2010) by incorporating the simultaneous allocation-sequencing of surgeries to both ORs and anesthesiologists. The closest mathematical model to GORPS is the work of Silva et al. (2015), which models an OR scheduling problem with multiple resources to maximize OR utilization. The authors use a time-based modeling paradigm (with OR and surgical time discretization of 30-minute time slots). Most time-based mathematical models either use a five-minute (Hashemi Doulabi et al., 2016) or a 15-minute (Marques et al., 2012) time discretization. Time discretization has some challenges and there is a trade-off

between the size of discretization and approximation quality. Smaller time discretization schemes lead to better objective function values with continuous surgical and OR times, but tractability of the problem reduces, which is due to the increased number of variables and constraints. All these studies (Denton et al., 2010; Rath et al., 2017; Silva et al., 2015) have developed daily OR schedules, while in this paper we present a weekly multi-resource constrained OR schedules that take into account OR downstream requirements for each surgery, surgery due-date, heterogeneous performances of members of resources, and lag times between the start and finish times of resources.

2.2 Solution methods for OR scheduling problems

The use of MIP and CP models are justifiable for either small OR scheduling problems that optimality can be achieved within a reasonable timeframe or for large-scale optimization problems that only a good integer feasible solution is required. As the optimization model grows in size, different problem-specific algorithms are required, including heuristics, meta-heuristics, two-stage sub-optimal heuristics, and optimal decomposition techniques. OR management decisions can be made sequentially, using sub-optimal sequential approaches (Begen and Queyranne, 2011; Castro and Marques, 2015; Fei et al., 2010; Gul et al., 2011; Jebali et al., 2006), or concurrently, using exact techniques such as mathematical programming models solved via existing optimization solvers (Hashemi Doulabi et al., 2016; Marques et al., 2012; Pham and Klinkert, 2008; Roshanaei et al., 2017a; Silva et al., 2015; Vijayakumar et al., 2013) and problem-specific decompositions (Batun et al., 2011; Hashemi Doulabi et al., 2016; Riise et al., 2016; Roshanaei et al., 2017a,b). Mathematical programming has been the most common approach to solving OR scheduling problems; however, recent studies have demonstrated that CP-based decomposition techniques are effective (Hashemi Doulabi et al., 2016). The simultaneous optimization of allocation-assignment-sequencing decisions in a single mathematical or CP model is beyond the capability of existing optimization solvers for realistically-sized problems (Hashemi Doulabi et al., 2016; Marques et al., 2012). To handle the intractability of OR scheduling problems, a wide variety of exact techniques have been developed, including branch-and-price (Belien and Demeulemeester, 2008; Cardoen et al., 2009; Fei et al., 2008; Hashemi Doulabi et al., 2016), Lagrangian relaxation (Augusto et al., 2010; Perdomo et al., 2006), and LBBD (Riise et al., 2016; Roshanaei et al., 2017a,b). None of these LBBD approaches can be directly used to solve GORPS due to the incorporation of new variables and constraints, which require a design of new Benders cuts. Additionally, none of these techniques use the type of resource-based decomposition scheme that we incorporate into the LBBDs that solve GORPS problem.

LBBD (Hooker, 2007; Hooker and Ottosson, 2003) approaches are recent exact techniques that have received significant attention in OR scheduling literature (Riise et al., 2016; Roshanaei et al., 2017a, 2020b, 2017b). Jebali et al. (2006) showed that sequencing SPs can be solved in two ways: (i) pure sequencing in which only binary sequencing variables are optimized given the fixed first-stage allocation variable values and (ii) sequencing with possible reallocation in which both allocation and sequencing variables are optimized. The advantage of the former approach is the fast computation of objective function value at the expense of slightly worse objective function value; while the second approach is more computationally burdensome, it is likely to provide a better objective function value. To the best of our knowledge, there is no study in the literature that takes advantage of these two sequence modeling paradigms to design effective Benders cuts to solve OR scheduling problems. A recent literature review on the advances in Benders decomposition approaches corroborates the novelty of the features that we incorporate into our LBBDs, including decomposition schemes, SP relaxations, Benders cuts, and variable-fixing cuts (Rahmaniani et al.,

2017). Additionally, to the best of our knowledge, there is no exact method in OR scheduling literature that simultaneously captures the heterogeneous performance of resources and downstream units of ORs.

3 Problem description and mathematical modeling

GORPS encompasses two general decision-making phases: (i) *planning* and (ii) *scheduling*. In the planning phase, patients are allocated to different days and their required number of resources is determined. In the scheduling phase, patients are assigned to allocated members of different resources and their sequence of surgeries within each resource member is determined. We consider all practical features contributing to the schedule, namely, time lags among resources, availability and eligibility of resources, and resource heterogeneity.

3.1 Problem description

Given a set of surgeries and resources (ORs, anesthetists, and surgeons) with heterogeneous performances and costs, we find the best allocation of surgeries and resources to days, assignment of resources to surgeries, and sequence of surgeries that leads to the minimum fixed- and over-time costs of these resources. We note that the per-surgery required amount of time from ORs and anesthetists is a function of surgical duration that itself depends largely on surgeons' performance. We thus only capture heterogeneity in surgeons' performance, because there is little variability in the performance of anesthetists. These resources perform their tasks on each surgery, taking into account predefined practical precedence and time lags. We elaborate on precedence among different members of different resources with an example. Table 1 shows the notation for GORPS.

Remark 1. *All models that we present in this study can capture performance heterogeneity of all resources; however, we only choose to capture heterogeneity in performance of surgeon resource as service times of all other resources depend on the performance of surgeons.*

Example of GORPS and dynamics among resources. Consider a surgery p with a given OR, anesthetist, and surgeon (Figure 1). Nurses in an OR can immediately start the process of preparing patients without any possible delay ($G_{p1} = 0$). Anesthetists can start administering anesthesia after the patient is prepared; their wait time is represented by G_{p2}. A surgeon can start the surgery with a time lag (G_{p3}) (after the anesthetist finishes his/her task) and stays in the OR until the end of surgery, which lasts for B_{p3}. After the start of the anesthesia, a surgeon cannot delay the start of a surgery by more than (G_{p3}). Thus, the service time of an anesthetist $B_{p2} = G_{p3} + B_{p3}$ and $W_p = 0$. Parameter E_p captures the cleaning time after each surgery. The total required time for each surgery in each OR, B_{p1}, is the summation of all the previous activities ($G_{p1} + G_{p2} + B_{p3} + E_p$). We define parameter W_p that allows nurse mangers (in consultation with surgeons) to specify the amount of time that anesthetists must spend on each surgery. For example, if $W_p = 0$, the anesthetist must be present in the OR until the surgical procedure is completed. If $W_p = 10$, the anesthetist can leave the OR 10, minutes before the surgeon finishes the surgical procedure. And, if $W_p = -10$, the anesthetist must leave the OR, 10 minutes after the surgeon finishes the surgery.

We note that all the features included in our models and methods and their values are obtained from our real dataset and from the interviews we have had with our clinical collaborators. The results of data analysis are presented in Appendix F. Here, we provide the list of some of these parameters (or modeling

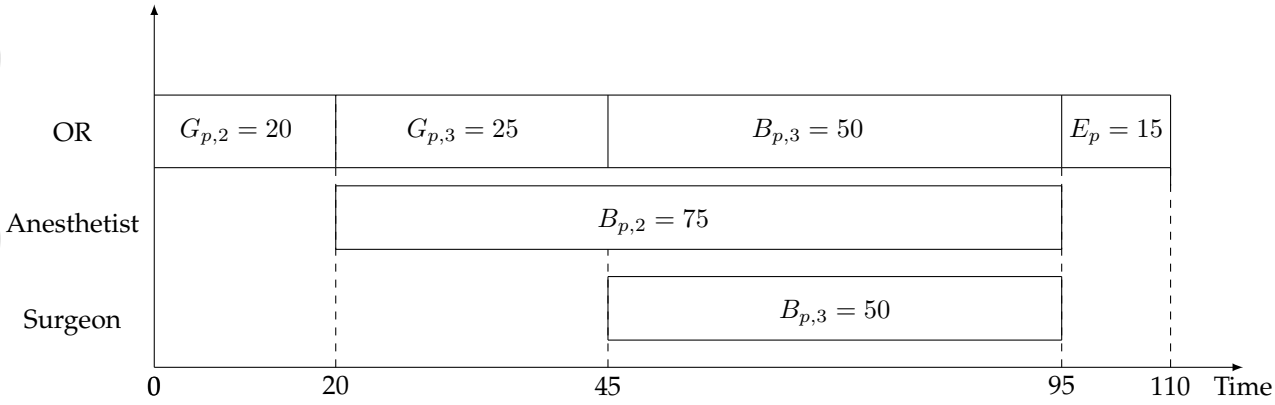


Figure 1: GORPS schematic representation: $G_{p1} = 0$: waiting time of an OR, $G_{p2} = 20$: waiting time of an anesthetist (patient preparation time), $G_{p3} = 25$: waiting time of a surgeon after anesthetist; $B_{p1} = 110$: time required for the surgery in any OR ($G_{p2} + G_{p3} + B_{p3} + E_p$), $B_{p2} = 75$: time required by an anesthetist, $B_{p3} = 50$: surgical time, $E_p = 15$: cleaning time after the surgery.

choices): (i) surgical and time lags among resources (Appendices F.1 and F.2), (ii) downstream units' route (Appendix F.3), (iii) heterogeneous performances of resources (Appendix F.4), (iv) open scheduling and resource sharing among specialties (Appendices F.5 and F.6), and (v) preferred working hours, regular and overtime (Appendix F.7), and the number of surgeries (Appendix F.8).

3.2 Mixed-integer programming models with continuous overtime

We develop MIP models using sequence-based, immediate-sequence-based, position-based, and time-based paradigms. We show that the first three sequence modeling paradigms yield optimal solutions, but the fourth one (time-based modeling) does not usually provide the optimal solution. Interestingly though, there seems to be a trend in the literature that shows time-based modeling has gained significant popularity (Hashemi Doulabi et al., 2016; Marques et al., 2012, 2014; Silva et al., 2015; Vijayakumar et al., 2013) compared to other potentially superior (computation and solution wise) modeling paradigms.

Investigating the performance of different mathematical models is paramount from both theoretical and practical perspectives. Practically speaking, some of these models, time-based models, in particular, yield significant OR under-utilization when their solutions are implemented in real clinical settings (see Castro and Marques (2015) for related discussion). This under-utilization will impact both hospitals and patients because the idle costs of surgeons and ORs could reach as high as $88.74 and $17.74 per minute, respectively (Batun et al., 2011). Due to recent trends in the adoption of such sub-optimal MIP approaches and their theoretical and practical impacts on OR scheduling practice, we conduct the first systematic comparisons among modeling paradigms that can be used to model sequencing decisions in an OR scheduling problem. Specifically, we wish to answer the following question: *"which of the existing sequence modeling paradigms yields a better trade-off between solution quality and computational tractability?."* The answer serves as a practical guide as which of these models is best suited for delivering and designing surgical services.

We also extend each model to treat the overtime of each member of each resource as a step function rather than as a continuous function (see Appendix C). These models are developed based on the following common assumption: master surgical scheduling has been done a priori (no resource overlapping exists among surgical specialties); therefore, only a single surgical specialty is considered (Denton et al., 2010;

Hashemi Doulabi et al., 2016). We first formulate GORPS using MIP and CP models with a continuous overtime function for each member of each resource. We show that GORPS can be formulated using different mathematical modeling paradigms. All these models capture heterogeneity of each resource member performances, because service times (preparation, anesthesia, and surgical) are resource-dependent. The notation for models is shown in Table 1. Below we present the most competitive modeling approach, sequenced-based modeling paradigm. (We chose this one for our models since it outperformed all others in our experiments. We present the other modeling approaches in the Appendix A).

Table 1: Notation for models

Sets:	
$\mathcal{P}$	Set of patients, $p \in \mathcal{P}$
$\mathcal{P}_h$	Set of patients requiring downstream unit $h \in \{1$=PACU, 2=ICU, 3=ward$\}$
$\mathcal{M}$	Set of resources, $m \in \mathcal{M}$ ($1 =$ ORs, $2 =$ anesthetists, $3 =$ surgeons)
$\mathcal{K}_m$	Set of members of resource m, $k \in \mathcal{K}_m$
$\mathcal{K}_{mp}$	Set of members of resource m eligible for patient p, $k \in \mathcal{K}_{mp}$
$\mathcal{D}$	Set of days, $d \in \mathcal{D}$
$\mathcal{D}_p$	Set of days on which surgery p can be performed (i.e., deadline)
$\mathcal{D}_{mk}$	Set of available days for kth member of resource m
Parameters:	
F_{mk}	Daily fixed cost of kth member of resource m
C_{mk}	Per unit overtime cost of using kth member of resource m
V_{dmk}	Maximum allowable daily overtime for kth member of resource m
B_{pmk}	The time that patient p requires from member k of resource m
G_{pm}	The time lag for resource m to start its operations on patient p after resource $m-1$
E_p	Cleaning time after patient p
W_p	The time lag between completion time of surgery p and anesthesia.
α	The number of available beds in PACU at each day.
β_d	The number of available ICU beds on day d
γ_d	The number of available Ward beds on day d
H_p	The length of stay of patient p in ICU
A_p	The length of stay of patient p in ward
T	Duration availability of each member of a resource

3.2.1 Sequence-based modeling paradigm

The sequence-based mixed-integer programming model, MIP_{SB}, investigates sequencing among any pair of surgeries that can be assigned to a member of a resource on each day. The precedence between any two arbitrary surgeries is not necessarily immediate in the MIP_{SB}. The decision variables for the MIP_{SB} are shown in Table 2. The order of m index implies inter-resource dependence, meaning resource member m must wait until members $1, \ldots, m-1$ have finished their tasks.

The MIP_{SB} with continuous overtime is as follows:

$$\min \quad \sum_{m\in\mathcal{M}} \sum_{k\in\mathcal{K}_m} \sum_{d\in\mathcal{D}_{mk}} \left(F_{mk} z_{dmk} + C_{mk} v_{dmk} \right) \tag{MIP$_{\text{SB}}$}$$

$$\text{s.t.} \quad \sum_{d\in\mathcal{D}_p} w_{pd} = 1 \qquad \forall p \in \mathcal{P} \tag{1}$$

Table 2: Variables for sequence-based modeling paradigm

Allocation	$w_{pd} \in \{0,1\}$	1 patient p is allocated to day d, 0 otherwise
	$z_{dmk} \in \{0,1\}$	1 if kth member of resource m is allocated to day d, 0 otherwise
Assignment	$x_{pmk} \in \{0,1\}$	1 if patient p is assigned to kth member of resource m, 0 otherwise
Sequencing	$y_{pp'} \in \{0,1\}$	1 if patient p is operated on after patient p' ($p > p'$), 0 otherwise
Scheduling	$s_{pm} \geq 0$	Starting time of resource m on patient p
	$c_{pm} \geq 0$	Completion time of resource m on patient p
	$f_{pdmk} \geq 0$	Finishing time of kth member of resource m for patient p on day d
	$v_{dmk} \geq 0$	Overtime amount of kth member of resource m on day d

$$\sum_{k \in \mathcal{K}_{mp}: \mathcal{D}_{mk} \cap \mathcal{D}_p \neq \emptyset} x_{pmk} = 1 \quad \forall p \in \mathcal{P}, m \in \mathcal{M} \tag{2}$$

$$x_{pmk} + w_{pd} \leq 1 + z_{dmk} \quad \forall p \in \mathcal{P}, m \in \mathcal{M}, k \in \mathcal{K}_{mp}, d \in \mathcal{D}_{mk} \cap \mathcal{D}_p \tag{3}$$

$$x_{pmk} \leq \sum_{d \in \mathcal{D}_{mk} \cap \mathcal{D}_p} w_{pd} \quad \forall p \in \mathcal{P}, m \in M, k \in \mathcal{K}_{mp} \tag{4}$$

$$c_{pm} \geq s_{pm} + B_{pmk} \cdot x_{pmk} \quad \forall p \in \mathcal{P}; m \in \mathcal{M}, k \in \mathcal{K}_{mp} \tag{5}$$

$$s_{pm} \geq c_{p'm} - M(5 - y_{pp'} - x_{pmk} - x_{p'mk} - w_{pd} - w_{p'd}) \quad \forall p > p', m \in \mathcal{M}, k \in \mathcal{K}_{mp}, d \in \mathcal{D}_p \cap \mathcal{D}_{p'} \cap \mathcal{D}_{mk} \tag{6}$$

$$s_{p'm} \geq c_{pm} - M(4 + y_{pp'} - x_{pmk} - x_{p'mk} - w_{pd} - w_{p'd}) \quad \forall p > p', m \in \mathcal{M}, k \in \mathcal{K}_{mp}, d \in \mathcal{D}_p \cap \mathcal{D}_{p'} \cap \mathcal{D}_{mk} \tag{7}$$

$$f_{pdmk} \geq c_{pm} - M(2 - x_{pmk} - w_{pd}) \quad \forall p \in \mathcal{P}, m \in \mathcal{M}, k \in \mathcal{K}_{mp}, d \in \mathcal{D}_p \cap \mathcal{D}_{mk} \tag{8}$$

$$s_{pm} \geq s_{p,m-1} + G_{pm} \quad \forall p \in \mathcal{P}, m \in \mathcal{M} \setminus 1 \tag{9}$$

$$c_{p1} \geq c_{p3} + E_p \quad \forall p \in \mathcal{P} \tag{10}$$

$$c_{p3} \leq c_{p2} + W_p \quad \forall p \in \mathcal{P} \tag{11}$$

$$V_{dmk} z_{dmk} \geq v_{dmk} \geq f_{pdmk} - T \quad \forall p \in \mathcal{P}, m \in \mathcal{M}, k \in \mathcal{K}_{mp}, d \in \mathcal{D}_p \cap \mathcal{D}_{mk} \tag{12}$$

$$\sum_{p \in \mathcal{P}_1: d \in \mathcal{D}_p} w_{pd} \leq \alpha \quad \forall d \in \mathcal{D} \tag{13}$$

$$\sum_{p \in \mathcal{P}_2: d \in \mathcal{D}_p} \sum_{d' = \max\{1, d - H_p + 1\}}^{d} w_{pd'} \leq \beta_d \quad \forall d \in \mathcal{D} \tag{14}$$

$$\sum_{p \in \{\mathcal{P}_3 \setminus \mathcal{P}_2\}: d \in \mathcal{D}_p} \sum_{d' = \max\{1, d - A_p + 1\}}^{d} w_{pd'} + \sum_{p \in \{\mathcal{P}_3 \cap \mathcal{P}_2: d \in \mathcal{D}_p\} | d - H_p \geq 1} \sum_{d' = \max\{1, d - H_p - A_p + 1\}}^{d - H_p} w_{pd'} \leq \gamma_d \quad \forall d \in \mathcal{D} \tag{15}$$

$$z_{dm,k-1} \geq z_{dmk} \quad \forall m = \{1,2\}, k \in \mathcal{K}_{mp}, d \in \mathcal{D}_{mk} \tag{16}$$

$$v_{dm,k-1} \geq v_{dmk} \quad \forall m = \{1,2\}, k \in \mathcal{K}_{mp}, d \in \mathcal{D}_{mk} \tag{17}$$

$$f_{pdmk}, s_{pm}, c_{pm}, v_{mdk} \geq 0 \quad \forall p \in \mathcal{P}, d \in \mathcal{D}, m \in \mathcal{M}, k \in \mathcal{K}_{mp} \tag{18}$$

$$x_{pmk}, z_{dmk}, w_{pd}, y_{pp'} \in \{0,1\} \quad \forall p > p', d \in \mathcal{D}, m \in \mathcal{M}, k \in \mathcal{K}_{mp} \tag{19}$$

MIP_{SB} minimizes fixed and variable (overtime) costs for each member of each resource. Each surgery is exactly assigned to only one of the days in the planning horizon by Constraint (1). Constraint (2) allocates patients to resources provided for each surgery. Constraint (3) assigns surgeries to different members of

different resource in each day only if they are used in that day. Constraint (4) ensures feasibility in terms of resource-day availability. Constraint (5) ensures that the difference between completion and starting time of a surgery for each member of a resource is at least as large as the time required from that member of that resource. Constraints (6) and (7) ensure that starting times of any pair of surgeries assigned to each member of each resource in each day do not overlap. Constraint (8) computes the finishing time of each surgery on each member of each resource, if the surgery is assigned to that member. The value of big M in Constraints (6) - (8) is identical and represents the maximum session length of an OR: regular OR time plus maximum allowable overtime for each OR, which in our case is 600 minutes. These values are the same in other MIP models and the LBBD method. Constraints (9) - (11) show the time lag among starting times of different resources on a surgery as explained by Figure 1. Constraint (12) captures the amount of overtime used for each member of a resource and ensures that the amount of allocated overtime does not exceed maximum allowable overtime, V_{dmk}. Constraints (13) - (15) ensure that the number of patients allocated to PACU, ICU, and ward beds does not exceed the daily number of beds, respectively. Valid inequalities (16) and (17) ensure that the regular time and overtime of the most cost-effective ORs and anesthetists are used first. The remaining Constraints (18) and (19) are non-negativity and binary constraints for the corresponding variables.

Other MIP models are presented in Appendix (A). A comparison among the parametric and numerical size of these MIPs in terms of number of variables and constraints is presented in Appendix (B).

3.2.2 Constraint programming (CP) paradigm

CP is another technique to solve scheduling problems and it has promising results for formulations with disjunctive (sequencing) constraints. CP also performs well in finding good integer solutions or just a single integer solution for optimization problems in a short amount of time. However, CP seems to be not as effective as conventional integer programming techniques in achieving optimality, but it is more effective in finding good-enough integer solutions. We use symbolic constraints, called global constraints (Laborie, 2009). For example, global constraint cp::AllDifferent($x_1, \ldots, x_n$) indicates that variables $x_1, \ldots, x_n$ must take unique values. Table 3 shows the global functions and constraints used in the CP model.

Table 3: Global functions and constraints used in the CP model

Functions:	
IntervalVar()	returns an interval variable.
BoolVar()	returns a binary variable.
EndOf(x)	returns the end of interval variable x.
PresenceOf(x)	returns the presence status of interval variable x.
Constraints:	
Alternative(x, y)	creates an alternative constraint between interval variable x and the set of interval variables y.
NoOvelap(y)	constrains a set of interval variables y not to overlap each other.
StartBeforeStart(x, y, a)	constrains the minimum delay between starts of two interval variables x and y to be at least a.
EndBeforeEnd(x, y, a)	constrains the minimum delay between the ends of two interval variables x and y to be at least a.

The CP model for GORPS is as follows:

$$\min \quad \text{Sum}_{d,m,k}\Big(\{F_{mk}\cdot z_{dmk}\}+\{C_{mk}\cdot \max_{p\in\mathcal{P}}\big(\max\{0,\text{EndOf}(\text{Task}_{pdmk})-T\}\big)\}\Big) \qquad \text{(CP model)}$$

$$\text{s.t.} \quad \text{Task}_{pdmk}=\text{IntervalVar}(B_{pmk},\text{Optional},\text{End=}[0,T+V_{dmk}]) \quad \forall p\in\mathcal{P}, m=3, k\in\mathcal{K}_{mp}, d\in\mathcal{D}_p\cap D_{mk} \tag{20}$$

$$\text{Task}_{pdmk}=\text{IntervalVar}(\text{Optional},\text{End=}[0,T+V_{dmk}]) \quad \forall p\in\mathcal{P}, m\in\mathcal{M}\setminus 3, k\in\mathcal{K}_{mp}, d\in\mathcal{D}_p\cap D_{mk} \tag{21}$$

$$w_{pd}, z_{dmk}=\text{BoolVar}() \quad \forall p\in\mathcal{P}, m\in\mathcal{M}, k\in\mathcal{K}_{mp}, d\in\mathcal{D}_p\cap D_{mk} \tag{22}$$

$$\text{Sum}\big(w_{pd}: d\in\mathcal{D}_p\big)=1 \quad \forall p\in\mathcal{P} \tag{23}$$

$$\text{Sum}\big(w_{pd'}: p\in\mathcal{P}_1|d\in\mathcal{D}_p\big)\le\alpha \quad \forall d\in\mathcal{D} \tag{24}$$

$$\text{Sum}\big(w_{pd'}: p\in\mathcal{P}_2|d\in\mathcal{D}_p,\max\{1,d-H_p+1\}\le d'\le d\big)\le\beta_d \quad \forall d\in\mathcal{D} \tag{25}$$

$$\text{Sum}\big(w_{pd'}: p\in\mathcal{P}_3\setminus\mathcal{P}_2|d\in\mathcal{D}_p,\max\{1,d-A_p+1\}\le d'\le d\big)+ \text{Sum}\big(w_{pd'}: p\{\mathcal{P}_3\cap\mathcal{P}_2\}|d-H_p\ge 1, d\in\mathcal{D}_p,\max\{1,d-H_p-A_p+1\}\le d'\le d-H_p\big)\le\gamma_d \quad \forall d\in\mathcal{D} \tag{26}$$

$$\text{PresenceOf}\big(\text{Task}_{pdmk}\big)\le w_{pd} \quad \forall p\in\mathcal{P}, m\in\mathcal{M}, k\in\mathcal{K}_{pm}, d\in\mathcal{D}_p\cap D_{mk} \tag{27}$$

$$\text{PresenceOf}\big(\text{Task}_{pdmk}\big)\le z_{dmk} \quad \forall p\in\mathcal{P}, m\in\mathcal{M}, k\in\mathcal{K}_{pm}, d\in\mathcal{D}_p\cap D_{mk} \tag{28}$$

$$\text{Alternative}\Big(\text{Task}^*_{pm},\big(\text{Task}_{pdmk}: k\in\mathcal{K}_{mp}, d\in\mathcal{D}_p\cap D_{mk}\big)\Big) \quad \forall p\in\mathcal{P}, m\in\mathcal{M} \tag{29}$$

$$\text{NoOverlap}\Big(\text{Task}_{pdmk}: p\in\mathcal{P}|k\in\mathcal{K}_{pm}, d\in\mathcal{D}_p\Big) \quad \forall m\in\mathcal{M}, k\in\mathcal{K}_m, d\in D_{mk} \tag{30}$$

$$\text{StartBeforeStart}\big(\text{Task}^*_{pm-1},\text{Task}^*_{pm},G_{pm}\big) \quad \forall p\in\mathcal{P}, m\in\mathcal{M}\setminus 1 \tag{31}$$

$$\text{EndBeforeEnd}\big(\text{Task}^*_{pM},\text{Task}^*_{p1},E_p\big) \quad \forall p\in\mathcal{P} \tag{32}$$

$$\text{EndOf}\big(\text{Task}^*_{pM}\big)\le\text{EndOf}\big(\text{Task}^*_{p2}\big)+W_p \quad \forall p\in\mathcal{P} \tag{33}$$

The objective function calculates the regular and overtime costs. Constraints (20) and (21) define the interval variable for each possible operation. Constraint (20) requires three input arguments: length of the interval (B_{pmk}), optionality, and possible finishing time of the interval. Constraint (22) defines Boolean variables for all possible allocations. Constraint (23) allocates the patient to days, similar to constraint (1). Constraints (24), (25), and (26) address downstream units, similar to constraints (13), (14), and (15), respectively. Constraint (27) removes inactive intervals for the days other than the day each patient is assigned to, similar to constraint (4). Constraint (28) determines which members of each resource are used, similar to constraint (3). Constraint (29) assigns patients to members of resources, similar to constraint (2). Constraint (30) specifies the interval of each member has no overlap, similar to constraints (6) and (7). Constraints (31), (32), and (33) ensures the time lag between resources, similar to constraints (9), (10), and (11), respectively.

3.3 Modeling discussion

We develop five models for GORPS problem using different paradigms, four of which are MIPs, and one of which is CP. The trade-off in these four MIP models is between branching efficiency (due to the lower number of binary variables) and bounding effectiveness (due to the inclusion of tighter LP relaxation). Branching efficiency leads to quick obtainment of an integer feasible solution, and bounding effectiveness

culminates in effective pruning of inferior nodes, resulting in faster convergence to optimality.

Common to all of our MIP models are the binary variables used to formulate the planning phase of GORPS and the continuous variables used to determine the start and finish times of surgeries for each member of each resource. The difference among models stems from assignment and sequencing decisions. MIP_{SB} and MIP_{PB} are able to handle assignment and sequencing decisions separately; therefore, they require fewer variables. MIP_{SB} has a three-indexed assignment variable and a two-indexed sequencing variable. These two variables are both three-indexed in the MIP_{PB}. The MIP_{ISB} and MIP_{TB} must integrate these two decisions, resulting in a five-indexed assignment-sequencing binary variables. The reason is that MIP_{ISB} and MIP_{TB} must define a separate sequence for each member of each resource on each day; therefore, each element appears as an index in the decision variable. While MIP_{SB} and MIP_{PB} define a general sequence and relaxes some sequences if two operations are done on different days or by different members of resources. Note that MIP_{PB} defines a sequence for each day because, in downstream units, we need to track the patient in each position of each day.

We have captured the parametric size dimensionality of these MIPs models and provided a numerical example to show how their sizes are impacted as we increase the number of patients and resources (Appendix B). Initial experiments demonstrated the absolute superiority of the MIP_{SB}.

4 Decomposition alternatives: an overview

We first present a brief description of LBBD and then present our decomposition approaches for GORPS.

4.1 LBBD

We develop LBBD techniques to efficiently solve models developed for GORPS. LBBD (Hooker, 2005, 2007; Hooker and Ottosson, 2003) is an exact technique that has proven its effectiveness in solving many large-scale combinatorial optimization problems. Significant computational improvements have been reported in several studies, including facility location (Fazel-Zarandi and Beck, 2012; Fazel-Zarandi et al., 2013), home healthcare planning and scheduling (Heching et al., 2019), operating room scheduling (Riise et al., 2016; Roshanaei et al., 2020a, 2017a), parallel machine scheduling with sequence-dependent setups (Tran et al., 2016), two-dimensional bin-packing (Pisinger and Sigurd, 2007), assembly line balancing (Naderi et al., 2018), multi-period network interdiction (Enayaty-Ahangar et al., 2018), integrated process planning and scheduling (Barzanji et al., 2019), and order acceptance and parallel machine scheduling (Naderi and Roshanaei, 2020) problems. Most of these LBBDs hybridize integer and constraint programming techniques and exploit their complementary strengths to ensure faster convergence.

LBBD decomposes the problem into two smaller and more computationally manageable models, an optimization master problem (MP) and one or more optimization (or feasibility/satisfaction) sub-problems (SPs) (Figure 2). SPs are used to either check feasibility of MP solutions (for feasibility SPs) or to find bounds (upper bound for minimization problem and lower bound for maximization problem) for MP solutions (if SPs are optimization problems). For infeasible and sub-optimal SPs, we develop Benders feasibility and optimality cuts, respectively. Unlike classical Benders decomposition, LBBD does not impose linear programming restriction on SPs and thus allows for the consideration of various combinatorial SPs. As a result, we cannot use the standard machinery (strong duality) used in the classical Benders decomposition to develop Benders cuts. The crux of an LBBD design is the development of valid and strong Benders cuts,

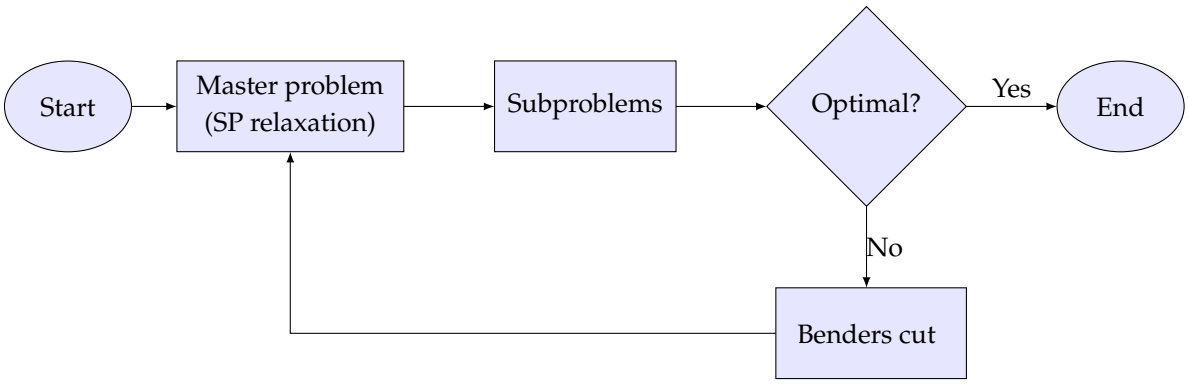


Figure 2: The general scheme of the LBBD

emanating from SP solutions (or statuses, including infeasibility) and communicating them back to the MP to generate a new solution.

4.2 Decomposition alternatives

There are multiple factors that impact an LBBD convergence: (i) the way the problem is decomposed (as we show in this paper with our new partitioning procedure in resource-based LBBD), (ii) the quality of the SP relaxations that are incorporated into the MP (Heching et al., 2019), (iii) the strength of Benders cuts (Hooker, 2007), and (iv) the choice of optimization approaches used to solve SPs (Fazel-Zarandi and Beck, 2012; Tran et al., 2016). Most of these factors are directly impacted by the way the problem is decomposed (factor (i)). This issue has not received enough attention in the literature as most studies have focused mostly on designing stronger SP relaxation and/or Benders cuts. We thus contribute to the LBBD literature by demonstrating that stronger SP relaxations and Benders cuts can be naturally devised, without using cut-strengthening techniques (as proposed in (Hooker, 2007), only as a result of the way a mathematical (or constraint) programming model is decomposed. We consider two general categories of decision variables and constraints (denoted without indices for simplicity) in GORPS:

1. **planning decision variables**, consisting of *allocation* variables that optimally allocate patients-to-days (denoted by $\mathbf{W}$) and resources-to-days (denoted by $\mathbf{Z}$) given the set of patients allocated to that day;
2. **scheduling decision variables**, consisting of (i) *assignment* variables that optimally assign allocated resources to each day to the set of scheduled patients in that day (denoted by $\mathbf{X}$) and (ii) *sequencing* variables (denoted by $\mathbf{Y}$) that order (and schedule) patients allocated to each member of each resource in each day.

All models developed in the previous section optimize GORPS decisions jointly (Figure 3). Depending on what combination of these variables and constraints we incorporate into MP and SPs, various LBBDs can be designed. We discuss possible ways to decompose our models and provide rationale behind each of these decomposition alternatives. Our decomposition methods differ in the way they partition variables and constraints in their MP and SPs, yielding LBBD variants of highly variable performances (see Figure 4). In most conventional decomposition algorithms, we optimize each set of a decision variable in either MP or SP. The alternative approach allows the entire set of already *globally* optimized variables in the MP to be *locally* optimized in SPs to achieve stronger bounds at the expense of more difficult SPs. As a result,

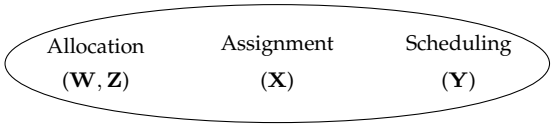


Figure 3: Classification of decisions in GORPS. MIP and CP models optimize all variables simultaneously.

the entire set of a decision variable may appear in both the MP and SPs. We show that we can decompose our models in such a way that strikes a balance between these two approaches.

We decompose GORPS into a weekly planning MP and multiple independent daily scheduling SPs. In general, the MP determines the optimal allocation of patients and resources for each day with the objective of minimizing the fixed cost of resources and achieving a relaxed value (lower bound) on the overtime amount for each member of each resource. Given the allocated sets of patients and resources, SPs optimally assign patients to resources and sequence patients to determine the minimum overtime amount for each member of each resource (upper bound). The connection between the MP and SPs is established via Benders optimality (and feasibility) cuts that allow the MP to trade off the unforeseen cost of SP overtimes due to sequencing variables and constraints with the cost of reallocating patients and resources to different days. We optimize the MP using a mathematical programming model (solved via CPLEX) and SPs using both mathematical (solved via CPLEX) and constraint (solved via CPOptimizer) programming model, and note that IP/CP decomposition methods are by far more popular than IP/IP decompositions (Fazel-Zarandi and Beck, 2012; Heching et al., 2019; Ku and Beck, 2016; Ku et al., 2014; Roshanaei et al., 2020a; Tran et al., 2016). We compare these decomposition to determine which scheme is best suited for GORPS.

We discuss four LBBD variants that are of varying MP and SP difficulties. In LBBD$_1$ (Figure 4(a)), we optimize the entire set of planning variables $\mathbf{W}$ and $\mathbf{Z}$ in the MP and scheduling variables $\mathbf{X}$ and $\mathbf{Y}$ in SPs. LBBD$_1$ possesses the easiest MP and the most difficult SPs. LBBD$_2$ (Figure 4(b)) transfers all assignment variables $\mathbf{X}$ from its SPs to its MP, resulting in a more difficult planning MP and much easier scheduling SPs (in fact, the easiest SPs among all LBBD variants). LBBD$_3$ (Figure 4(c)) optimizes variables $\mathbf{W}$, $\mathbf{Z}$ and $\mathbf{X}$ in its MP (the same difficulty as the MP in LBBD$_2$), but allows for re-optimization of variable $\mathbf{X}$ in its SPs (the same difficulty as SPs in LBBD$_1$). Note that these three LBBD variants (that are similar to many existing LBBDs in the literature) optimize or re-optimize the entire set of a variable in their MPs or SPs. We use as an example "assignment variables" to clarify what we mean by the entire set of a variable. In LBBDs discussed earlier, regardless of where the intended variable is optimized/re-optimized, we determine the optimal assignment of *all* resources, i.e., surgeons, ORs, and anesthetists for each scheduled surgery in each day.

Unlike LBBD$_3$ that optimizes the entire set of assignment variable $\mathbf{X}$ in its MP, LBBD$_4$ (Figure 4(d)) partially optimizes this variable in its MP (only for surgeons, denoted by $\mathbf{X}_3$) and allows for optimization/re-optimization of the entire set of variables $\mathbf{X}$ (including variable $\mathbf{X}_3$) and $\mathbf{Y}$ in its SPs. The MP in LBBD$_4$ is easier than those of LBBD$_2$ and LBBD$_3$, but more difficult than that of LBBD$_1$. On the other hand, LBBD$_4$ possesses SPs that are as difficult as those of LBBD$_1$ and LBBD$_3$, but more difficult than that of LBBD$_2$. The non-trivial question that naturally arises in such a situation is which of these algorithms will yield a better schedule (low cost) within a reasonable clinical time frame?, which we will address in this study.

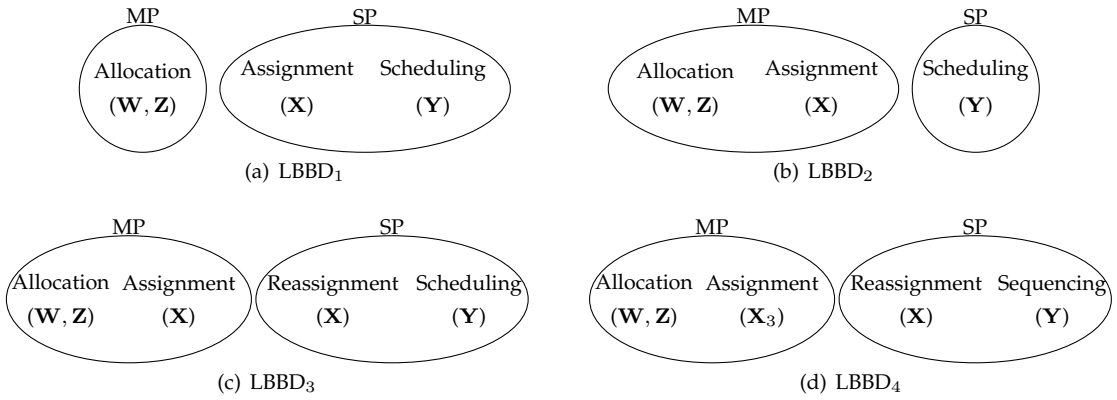


Figure 4: Structural decomposition possibilities in the original models

4.3 Advantages and disadvantages of LBBDs

We discuss advantages and disadvantages of each of our four decomposition alternatives. We provide a few remarks that we deem are essential for better understanding of our LBBDs.

Remark 2. *In all our LBBDs, MP is responsible for determining the global lower bound, whereas SPs are responsible for finding candidate solutions that constitute global upper bounds for each MP solution at each LBBD iteration.*

For our GORPS problem, we find the following factors substantially influence the performance of LBBD:

1. **Balance of MP and SPs' computational difficulty**: After giving due consideration to various partitioning structures in our MIP/CP model, we discover that those partitioning procedures that balance MP and SPs difficulty are likely to yield better solution quality. This finding is unlike the existing trend in the literature that strives to tractably solve SPs with, say, polynomial techniques. We show there exists a trade-off between SP difficulty and solution quality; as such, the fastest SP does not necessarily yield the best performance.

2. **SP relaxation**: In most LBBD algorithm designs (especially textbook implementation of classical and logic-based Benders algorithms), the MP and SPs optimize variables exclusive to them. Such a disconnect between the variables that the MP optimizes and those of SPs has been resolved by the concept of SP relaxation (Heching et al., 2019; Hooker, 2007). (Remark 4 defines SP relaxation.) In the context of GORPS, the incorporation of SP relaxation into the MP causes the MP to produce solutions whose estimated overtime values are closer to actual overtime values determined by SPs.

3. **SP flexibility**: SPs in LBBDs can have varying levels of flexibility. By flexibility we mean to allow SPs to optimize more variables. If most variables are optimized in the MP and few of them are left to be optimized in the SPs, we have a rigid SP. Rigid SPs constrained by the MP-optimized values are more likely to be infeasible than flexible SPs that have the advantage of optimizing more variables.

4. **Benders cut and symmetry**: A common way to allow MP to generate tighter lower bounds and solutions for SPs is to include as many variables and constraints as possible in the MP. Although this approach causes MP to achieve tighter bounds, it gives rise to the issue of MP difficulty and even worse *symmetrical solutions*, i.e., MP might find multiple similar solutions that yield the same objective function value. Multitude of such symmetrical solutions in MP feasible region hinders LBBDs'

convergences even though we have strong LBBD cuts because the MP needs to exhaustively evaluate all these symmetrical solutions (see Table 4).

Remark 3. *Incorporating allocation variables into MP and scheduling variables (assignment and sequencing) into SP can be immediately considered as the best option for LBBD design. The success in our LBBD design, however, depends on where we optimize assignment variables (that causes symmetry in our models) as they impact all the four factors that we discussed above.*

We discuss four possible ways that assignment variables can be optimized in our LBBDs: (i) the entire set of assignment variables for all resources are optimized in the MP ($LBBD_1$), (ii) the entire set of assignment variables for all resources are optimized in SPs ($LBBD_2$), (iii) the entire set of assignment variables for all resources are optimized in the MP (in conjunction with other allocation variables) and SPs (in conjunction with sequencing variables) ($LBBD_3$), and (iv) a partial subset of assignment variables (for some resources) in MP and its entire assignment variables are re-optimized in SPs (in conjunction with sequencing variables) ($LBBD_4$). See Table 4 for advantages and disadvantages of each LBBD variant. Initial experiments revealed that $LBBD_4$ is superior to other LBBDs. We thus continue presenting only the details of $LBBD_4$.

Table 4: Comparison of LBBDs' strengths based on their structural constituents. Weak, strong, and mid SP relaxations are based on total workload, individual workload, and the hybrid of total/individual workloads, respectively.

LBBD	MP			SP		Benders cuts		Description
	Variables	SP relaxation	Difficulty	Variables	Flexibility	Variables	Symmetry	
$LBBD_1$	**W, Z**	Weak	Easy	**X,Y**	High	**W, Z**	Low	Low chance of SP feasibility as MP overloads resources due to weak SP relaxation.
$LBBD_2$	**W, Z, X**	Strong	Hard	**Y**	Low	**X**	High	Low chance of SP feasibility due to SP inflexibility; MP is also difficult.
$LBBD_3$	**W, Z, X**	Strong	Hard	**X, Y**	High	**W, Z**	Low	MP is difficult; higher chance of SP feasibility due to SP flexibility.
$LBBD_4$	**W, Z, X_3**	Mid	Mid	**X, Y**	High	**W, Z**	Low	Balanced MP and SP difficulty; higher chance of SP feasibility.

5 $LBBD_4$: a computationally balanced approach to LBBD design

In the previous section, we provided rationale behind the development of each of our LBBD decomposition variants. To avoid lengthy expositions of LBBDs and because of the absolute superiority of $LBBD_4$ compared to other LBBDs (see results in Section 6), we only present $LBBD_4$. We thus present the MP, SPs, SP relaxations with their linearization schemes, and Benders cuts for this LBBD. We discuss that SPs can be solved using both mathematical (which is based on MIP_{SB}) and constraint programming models.

5.1 Joint allocation/assignment MP (AAMP) with SP relaxation

$LBBD_4$ optimizes allocation variables $\mathbf{W}$ and $\mathbf{Z}$ as well as only surgeon-to-patient assignment variables $\mathbf{X}_3$ in its MP, leaving scheduling variables, consisting of the entire set of $\mathbf{X}$ and $\mathbf{Y}$, to be re-optimized and optimized in its SPs, respectively. We enhance the MP via our novel SP relaxations (see Section 5.2). The

mathematical model of the MP is as follows:

$$\text{minimize} \quad \sum_{d\in\mathcal{D}}\sum_{m\in\mathcal{M}}\sum_{k\in\mathcal{K}_m}\Big(F_{mk}z_{dmk}+C_{mk}v_{dmk}\Big) \tag{AAMP}$$

subject to Constraints (1)-(4), (13)-(17)

SP relaxation (see Section 5.2)

$$0 \le v_{dmk} \le V_{dmk}\cdot z_{dmk} \qquad \forall m\in\mathcal{M}, k\in\mathcal{K}_m, d\in\mathcal{D}_{mk} \tag{34}$$

$$x_{pmk}, w_{pd}, z_{dmk} \in \{0,1\} \qquad \forall p\in\mathcal{P}; d\in\mathcal{D}; m\in\mathcal{M}; k\in\mathcal{K}_m \tag{35}$$

Constraint (34) ensures that the allocated amount of overtime to each member of each resource does not exceed the maximum allowable overtime for that resource.

Remark 4. *"SP relaxation" is a set of surrogate constraints that estimate the objective function of SP using decision variables of MP. The inclusion of these variables and constraints in the MP leads to more accurate MP solutions.*

5.2 SP Relaxations

Sequencing variables, $\mathbf{Y}$, and the constraints embracing these variables are notorious for their intractability and poor LP relaxation due to the disjunction concept (big M) that they use to avoid any overlap in the starting times of surgeries for different resources (Naderi and Ruiz, 2010). A workaround is to design surrogate constraints that are able to function, however weakly, in lieu of these sequencing constraints. These surrogate constraints are called relaxations in the LBBD literature. The incorporation of scheduling relaxations into the MP increases solution quality and reduces likelihood of MP-generated solutions being rejected in SPs. Additionally, it eliminates the need for big M constraints that are used to avoid surgery overlapping. Last but not least, it also strengthens MP to generate tighter global lower bounds.

The AAMP objective function consists of two parts, fixed and overtime costs. The fixed cost is a function of allocation variable $\mathbf{Z}$ that is obtained in the AAMP. The overtime cost is a function of scheduling variable $\mathbf{Y}$ that is obtained by SPs after the AAMP variables are optimized. Note that at the first iteration of the AAMP, there is no constraint regarding overtime calculation and thus its overtime value is zero, which is a loose lower bound. In this regard, we need to use a set of constraints in the AAMP that can estimate the overtime cost using variables in the AAMP. The tightness of the overtime estimation depends on what variables we incorporate into the AAMP. If we only include allocation variable $\mathbf{Z}$, then, the estimation is not going to be desirably tight (see *total* workload relaxation in Section 5.2.1), whereas if we include both allocation variable $\mathbf{Y}$ and assignment variable $\mathbf{X}$ in the AAMP, the estimation is much tighter (see *individual* workload relaxation in Section 5.2.1). Note that these relaxations provide only lower bounds on the actual amount of overtime that is determined in the SPs. Assuming identical computational efforts to derive these relaxations, tighter lower bound values culminate in faster LBBD convergence.

We develop two novel relaxations with varying tightness and computational efficiencies. The strength of relaxations varies depending on whether they capture heterogeneous performances of resources or not. Performance-based relaxations are able to determine a patient's surgical length in an OR and also anesthesia time, leading to a more realistic estimation of costs. In our study, tighter relaxations determine surgeon-to-patient assignments ($\mathbf{X}_3$) as they help compute the amount of time that other resources require to spend on each surgery. Thus, we categorize scheduling SP relaxations into two classes (i) *total workload estimation* (in constraint (37)) and (ii) *individual workload estimation* (in constraint (38)). To present our SP relaxations

Table 5: Auxiliary parameters and variables required for the SP relaxations in the MP

Parameters:	
G'_m	The minimum time lag for a member of resource m to start work, i.e., $G'_1 = 0, \quad G'_2 = \min_{p\in\mathcal{P}}\{G_{p1}\}, \quad G'_3 = G'_2 + \min_{p\in\mathcal{P}}\{G_{p2}\},$
Q'_{pm}	The extra length of stay of resource m for patient p, i.e., $Q'_{p1} = G_{p2} + G_{p3} + E_p, \quad Q'_{p2} = G_{p3}, \quad Q'_{p3} = 0,$
Variables:	
q^3_{pdk}	The length of stay of the kth member of resource 3 for patient p on day d
q^1_{pdm}	The length of stay of a member of resource $m = 1, 2$ for patient p on day d

(lower bound), we define a new set of notation in Table 5.

5.2.1 Lower bound of determining workloads

Our lower bound (Equation (36)) extends the lower bound of Denton et al. (2010) (proposed for a single-resource operating room allocation problem) in a variety of ways by considering (i) a limit on the use of overtime for resources, (ii) time lags among starting times of each member of each resource on each surgery (i.e., dynamics among different resources working on the same surgery), and (iii) heterogeneous performances of resources. Our lower bound is:

Proposition 1. *A lower bound on the number of members of a resource m on a given day d is calculated as follows:*

$$L_{dm} = \left\lceil \frac{\sum_{p\in\hat{\mathcal{P}}_d^{(i)}} B_{p3} + Q'_{pm}}{T - G'_m + V} \right\rceil \qquad \forall d, m, \tag{36}$$

where B_{p3} is the surgical time of patient p and V is the maximum allowable overtime.

This lower bound is general and can be incorporated in two different ways as total and/or individual workload relaxations in LBBDs (Figure 5). These relaxation variants vary in strength and complexity. The choice hings on partitioning procedure used in the LBBD; specifically, it depends on which set of assignment variables we include in the MP of the LBBD. If the surgeon-to-patient assignment is made in the MP, the exact value of B_{p3} is the surgical duration of that patient; otherwise, we must set $B_{p3} = \min_{k\in\mathcal{K}_m}\{B_{pmk}\}$, which means the minimum surgical time of a surgery type on the entire of set of eligible surgeons who can operate on that surgery. We use the *total workload relaxation* (LB_1) for a resource when the assignment variables of the resource is not included in the MP and *individual workload relaxation* (LB_2) when the assignment variables of the resource are included in the MP. Figure 5 illustrates how these two LB variants are used in different LBBDs.

Total workload relaxation (LB_1). We transform the lower bound in Equation (36) into the following *non-linear* constraint and add it to the MP:

$$\sum_{k\in\mathcal{K}_{pm}:d\in\mathcal{D}_{mk}} \overbrace{\left(\overbrace{(T - G'_m)z_{dmk}}^{\text{Regular time}} + \overbrace{v_{dmk}}^{\text{Overtime}}\right)}^{\text{Availability time of one member of resource } m} \geq \sum_{p\in\mathcal{P}:d\in\mathcal{D}_p} \overbrace{\left(Q'_{pm} + \overbrace{\sum_{k\in\mathcal{K}_{p3}:d\in\mathcal{D}_{3k}} B_{p3k}\cdot x_{p3k}}^{B_{p3}}\right)\cdot w_{pd}}^{\text{The estimated workload of patient } p \text{ for resource } m \text{ on day } d}$$
$$\forall m \in \mathcal{M}, d \in \mathcal{D}. \tag{37}$$

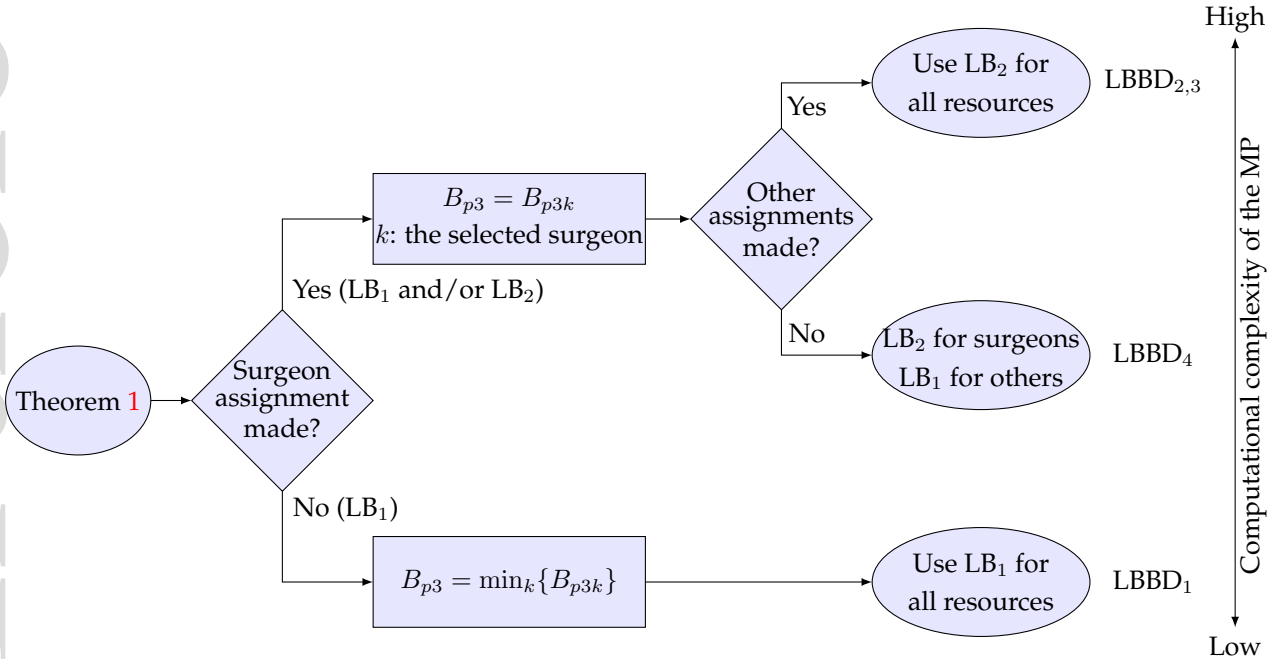


Figure 5: Proposition 1 can be implemented in two different ways for a resource: Total workload (LB_1: Inequality (37)) and individual workload (LB_2: Inequality (38)).

The right hand side of LB_1 in Inequality (37) is a lower bound for the total workload of resource m on day d. There are two ways to use LB_1. First, we can remove all assignment variables and work with $B_{p3} = \min_{k \in \mathcal{K}_3} \{B_{p3k}\}$. Second, we can include assignment of surgeons-to-patients in the MP to calculate the individual workload of surgeons $B_{p3} = B_{p3k}$ (where k is the selected surgeon) and total workload for other resources. Since the service time of other resources depends on the surgical duration that itself depends on the surgeons' performance, the assignment of the surgeon for each surgery can significantly enhance the accuracy of the estimation for total workload of other resources.

Remark 5. *The disadvantage of removing all assignment variables is that Inequality (37) provides a weak lower bound for GORPS. The advantage of removing all assignment variables is that Inequality (37) remains linear and hence tractable. We present a novel linearization scheme for variants that optimize assignment variables in their AAMP (and their associated LBs) in Section 5.2.2.*

Individual workload relaxation (LB_2). We use LB_2 for those resources whose assignment decisions are optimized in the AAMP. LB_2 enhances LB_1 by utilizing the knowledge acquired from these assignment decisions in the AAMP to estimate more accurate workloads. LB_2 is:

$$\overbrace{(T - G'_m)}^{\text{Regular time}} + \overbrace{v_{dmk}}^{\text{Overtime}} \geq \sum_{p \in P} \overbrace{\left(\overbrace{\left(\sum_{k' \in \mathcal{K}_{p3}} B_{p3k'} \cdot x_{p3k'} \right)}^{B_{p3}} + Q'_{pm} \right) x_{pmk} \cdot w_{pd}}^{\text{The estimated workload of patient } p \text{ for an individual member}} \quad \forall m \in \mathcal{M}, k \in \mathcal{K}_m, d \in \mathcal{D}_{mk}. \quad (38)$$

Non-linear Inequality (38) provides a much stronger lower bound than LB_1, but causes substantial computational burden to the AAMP. We present a novel linearization procedure for Inequality (38) in Section 5.2.2.

Table 6: Notation used in RSSPs. The index i indicates the AAMP solution at iteration i.

Notation	Definition
$\hat{\mathcal{P}}_d^{(i)}$	Set of patients allocated to day d in the AAMP solution at iteration i
$\hat{\mathcal{K}}_{dm}^{(i)}$	Set of members of resource m assigned to day d in the AAMP solution at iteration i
$\hat{V}_d^{(i)}$	Cost of overtime on day d determined by AAMP at iteration i for $\hat{\mathcal{K}}_{dm}^{(i)}$ given $\hat{\mathcal{P}}_d^{(i)}$
$\bar{V}_d^{(i)}$	Cost of overtime on day d determined by RSSP at iteration i for $\hat{\mathcal{K}}_{dm}^{(i)}$ given $\hat{\mathcal{P}}_d^{(i)}$
$\bar{\mathcal{D}}_{\text{sub-optimal}}^{(i)}$	Set of sub-optimal RSSPs
$\bar{\mathcal{D}}_{\text{infeasible}}^{(i)}$	Set of infeasible RSSPs

5.2.2 Linearization scheme for Inequalities (37) and (38)

We define a three-indexed auxiliary *continuous* variable $q^3_{pdk} \geq 0$ (see the definition in Table 5) to linearly calculate the individual workload for resource 3 (surgeons) in Inequality (38) and another three-indexed auxiliary continuous variable $q^1_{pdm} \geq 0$ to calculate the total workload for resources 1 and 2 (ORs and anesthetists) in Inequality (37). To do so, we use Constraint sets (39) - (46):

$$q^3_{pdk} \leq (T+V)w_{pd} \qquad \forall m=3, p\in\mathcal{P}, k\in\mathcal{K}_{pm}, d\in\mathcal{D}_p\cap\mathcal{D}_{mk}, \tag{39}$$

$$q^1_{pdm} \leq (T+V)w_{pd} \qquad \forall m=\{1,2\}, p\in\mathcal{P}, d\in\mathcal{D}_p, \tag{40}$$

$$q^3_{pdk} \leq (T+V)x_{pmk} \qquad \forall m=3, p\in\mathcal{P}, k\in\mathcal{K}_{pm}, d\in\mathcal{D}_p\cap\mathcal{D}_{mk}, \tag{41}$$

$$\sum_{d\in\mathcal{D}_p\cap\mathcal{D}_{mk}} q^3_{pdk} = B_{pmk}x_{pmk} \qquad \forall m=3, p\in\mathcal{P}, k\in\mathcal{K}_{pm}, \tag{42}$$

$$\sum_{d\in\mathcal{D}_p} q^1_{pdm} = Q'_{pm} + \sum_{k\in\mathcal{K}_{p3}}\sum_{d\in\mathcal{D}_p\cap\mathcal{D}_{3k}} q^3_{pdk} \qquad \forall m=\{1,2\}, p\in\mathcal{P}, \tag{43}$$

$$v_{dmk} \geq \sum_{p\in\mathcal{P}:k\in\mathcal{K}_{pm}, d\in\mathcal{D}_p} q^3_{pdk} - (T-G'_m) \qquad \forall m=3, k\in\mathcal{K}_m, d\in\mathcal{D}_{mk}, \tag{44}$$

$$\sum_{k\in\mathcal{K}_{pm}} \Big((T-G'_m)z_{dmk} + v_{dmk}\Big) \geq \sum_{p\in\mathcal{P}:d\in\mathcal{D}_p} q^1_{pdm} \qquad \forall m=\{1,2\}, d\in\mathcal{D}, \tag{45}$$

$$q^3_{pdk}, q^1_{pdm} \geq 0 \qquad \forall m\in\mathcal{M}, p\in\mathcal{P}, k\in\mathcal{K}_{pm}, d\in\mathcal{D}_p\cap\mathcal{D}_{mk}. \tag{46}$$

Constraints (39) and (41) limit the workload of each surgery p to the day and the member of each resource it is assigned to. Constraint (42) determines the workload for members of resource 3. Constraint (43) specifies the workload for resources 1 and 2. Constraint (44) estimates the overtime using LB_2. Constraint (46) is the non-negativity of the auxiliary variables.

AAMP output. Having incorporated constraints (39) - (46) into the AAMP, we solve the AAMP that minimizes the total fixed and overtime costs. The AAMP solution at each iteration i yields a set of patients ($\hat{\mathcal{P}}_d^{(i)}$ using variable w_{pd}) and resource members ($\hat{\mathcal{K}}_{dm}^{(i)}$ using variable z_{dmk}) for each day (see definition in Table 6). The AAMP also assigns a subset of patients ($\hat{\mathcal{P}}_d^{(i)}$) to each surgeon ($\hat{\mathcal{K}}_{d(m=3)}^{(i)}$ using variable x_{p3k}), which is later re-optimized (finalized) in RSSPs.

5.3 Re-assignment/scheduling sub-problems (RSSPs)

Each RSSP, given $\hat{\mathcal{P}}_d^{(i)}$ and $\hat{\mathcal{K}}_{dm}^{(i)}$, constructs a daily optimal OR schedule that minimizes overtime for each member of each allocated resource to that day. Constructing an optimal schedule for LBBD_4 entails final-

izing two additional decisions: (i) assignment of resources to surgeries using variable $x_{pmk} \in \{0,1\}$ and (ii) sequencing/scheduling of patients within the list of each resource in each day. For clarity of exposition, we differentiate between *optimize* and *finalize* because the MP may have optimized some of the assignment variables, but still allows for the re-optimization (finalization) of these variables in RSSPs. As such, we call our SPs "re-assignment/scheduling SP (RSSP)".

The set of resources and patients allocated to each day is unique (RSSPs receive unique input from the AAMP), i.e., the determination of overtime for each member of each resource in each day can be made independently of that of other days. We thus construct an RSSP for each day to which at least one patient has been allocated. As stated earlier, RSSPs can be formulated using mathematical and constraint programming models. We present the MIP model for our RSSP and provide the equivalent CP in Appendix D. The generic RSSP-MIP model for LBBD$_4$ is given as follows:

$$\text{minimize} \quad \sum_{m\in\mathcal{M}} \sum_{k\in\hat{\mathcal{K}}_{dm}^{(i)}} C_{mk} v_{mk} \qquad \text{(RSSP-MIP)}$$

$$\text{subject to:} \quad s_{pm} \geq s_{p,m-1} + G_{pm} \qquad \forall p \in \hat{\mathcal{P}}_d^{(i)}, m \in \mathcal{M} \setminus 1 \qquad (47)$$

$$s_{p3} \leq s_{p2} + W_p \qquad \forall p \in \hat{\mathcal{P}}_d^{(i)} \qquad (48)$$

$$c_{pm} \geq s_{pm} + \sum_{k\in\hat{\mathcal{K}}_{dm}^{(i)}\cap\mathcal{K}_{mp}} B_{pmk} \cdot x_{pmk} \qquad \forall m \in \mathcal{M}, p \in \hat{\mathcal{P}}_d^{(i)} \qquad (49)$$

$$s_{pm} \geq c_{p'm} - M(3 - y_{pp'} - x_{pmk} - x_{p'mk}) \qquad \forall (p,p') \in \hat{\mathcal{P}}_d^{(i)} : p > p', m \in \mathcal{M}, k \in \hat{\mathcal{K}}_{dm}^{(i)} \cap \mathcal{K}_{mp} \cap \mathcal{K}_{mp'} \qquad (50)$$

$$s_{p'm} \geq c_{pm} - M(2 + y_{pp'} - x_{pmk} - x_{p'mk}) \qquad \forall (p,p') \in \hat{\mathcal{P}}_d^{(i)} : p > p', m \in \mathcal{M}, k \in \hat{\mathcal{K}}_{dm}^{(i)} \cap \mathcal{K}_{mp} \cap \mathcal{K}_{mp'} \qquad (51)$$

$$f_{pmk} \geq c_{pm} - M(1 - x_{pmk}) \qquad \forall p \in \hat{\mathcal{P}}_d^{(i)}, m \in \mathcal{M}, k \in \hat{\mathcal{K}}_{dm}^{(i)} \cap \mathcal{K}_{mp} \qquad (52)$$

$$V_{mk} \geq v_{mk} \geq f_{pmk} - T \qquad \forall p \in \hat{\mathcal{P}}_d^{(i)}, m \in \mathcal{M}, k \in \hat{\mathcal{K}}_{dm}^{(i)} \cap \mathcal{K}_{mp}, \qquad (53)$$

$$f_{pmk}, s_{pm}, c_{pm}, v_{mk} \geq 0 \qquad \forall p \in \hat{\mathcal{P}}_d^{(i)}, m \in \mathcal{M}, k \in \hat{\mathcal{K}}_{dm}^{(i)} \qquad (54)$$

$$x_{pmk}, y_{pp'} \in \{0,1\} \qquad \forall (p,p') \in \hat{\mathcal{P}}_d^{(i)} : p > p', m \in \mathcal{M}, k \in \hat{\mathcal{K}}_{dm}^{(i)} \qquad (55)$$

The above constraints are similar to constraints in model MIP$_{SB}$ given optimal values of variables $\mathbf{W}$ and $\mathbf{Z}$ in AAMP.

5.4 Optimality and Benders cuts

We present optimality condition and how to update global bounds. We discuss conditions under which we develop Benders feasibility and optimality cuts.

5.4.1 Optimality

We update global bounds at each iteration i after we solve the AAMP and RSSPs. The AAMP and RSSPs are responsible for determining global LB and UB, respectively. That is, we use the AAMP optimal objective function value at iteration i to update the global LB at iteration i (denoted by LB$^{(i)}$). Note that the AAMP objective function consists of two components: fixed ($\hat{Z}_d^{(i)}$) and overtime ($\hat{V}_d^{(i)}$) costs per day d. We use the sum of RSSPs' daily overtime at iteration i (denoted by $\bar{V}_d^{(i)}$) plus $\hat{Z}_d^{(i)}$ to update the global UB at iteration i

(denoted by $LB^{(i)}$). Intuitively, $\sum_d(\bar{V}_d^{(i)} + \bar{Z}_d^{(i)}) \geq LB^{(i)}$ because at each iteration, we add cuts that tighten the AAMP feasible region. On the other hand, $\sum_d(\bar{V}_d^{(i)} + \hat{Z}_d^{(i)})$ is a candidate for the global UB. Note that we may have $\sum_d(\bar{V}_d^{(i)} + \hat{Z}_d^{(i)}) > UB^{(i)}$, that is, we may not improve global UB at each iteration. In all algorithms that we test, we allow for the premature stopping of the AAMP—this choice of implementation resembles the case that we stop the original MIP after a certain timelimit in which case the AAMP's LB and UB may not converge. In this case, the LB of the AAMP is still a valid global LB for the problem. At any LBBD iteration, if LBBD is not optimal (i.e., $\text{UB}^{(i)} > \text{LB}^{(i)}$), we develop Benders feasibility and optimality cuts. For each infeasible SP, we add one no-good Benders feasibility cut. For each feasible but sub-optimal SP, we add one Benders optimality cut.

5.4.2 Benders cuts

Logic-based Benders cuts are corrective (feedback) mechanisms that are cast as valid inequalities, incorporating remedial strategies into the AAMP to break infeasibility and/or sub-optimality of its solutions. Benders cuts are developed based on variables whose values that are finalized in the AAMP; therefore, they vary based on the AAMP type. We develop Benders feasibility and optimality cuts if any AAMP-optimized solution leads to an infeasible or sub-optimal RSSP, respectively. An RSSP is sub-optimal if $\hat{V}_d^{(i)} < \bar{V}_d^{(i)}$. Each valid Benders cut that is added to the AAMP must satisfy two properties (Chu and Xia, 2004): (i) It must cut off the current infeasible or sub-optimal solution from the AAMP feasible region and (ii) It must not cut off any other globally integer feasible solution from the AAMP feasible region.

Remark 6. *We develop logic-based Benders cuts based on the entire set or a partial subset AAMP variables. Benders cuts for LBBD$_4$ must include only variables whose final values are determined in the AAMP (i.e.,* $\mathbf{W}$ *and* $\mathbf{Z}$*) and not* $\mathbf{X}_3$ *whose final values are determined in RSSPs. The linking variable between the AAMP and RSSPs, overtime variable, is also included in the Benders optimality cut that is mostly bounded from below by* $\bar{V}_d^{(i)}$.

Benders feasibility cuts. If the AAMP solution in any LBBD results in infeasible RSSPs, we develop a Benders feasibility cut. The sole source of infeasibility in RSSPs is the boundedness of daily overtime amount that each member of each resource can use. The existence of sequencing constraints expands the high-utilization schedule generated by the AAMP, leading to infeasibility because the AAMP-allocated availability times may not be sufficient for at least one of resources. Essentially, the Benders feasibility cut rules out the AAMP optimum (or any integer feasible solution if AAMP is stopped before optimality) in the previous iteration and forces the AAMP to generate a new solution. The Benders feasibility cut for LBBD$_4$ is as follows:

$$\sum_{p \in \hat{\mathcal{P}}_d^{(i)}} (1 - w_{pd}) + \sum_{m \in \mathcal{M}} \sum_{k \in \left\{\mathcal{K} \setminus \hat{\mathcal{K}}_{dm}^{(i)}\right\}} z_{dmk} \geq 1 \qquad \forall d \in \bar{\mathcal{D}}_{\text{infeasible}}^{(i)}, \tag{56}$$

where $\bar{\mathcal{D}}_{\text{infeasible}}^{(i)}$ is the set of all infeasible RSSPs. This cut alters the combination of patients and/or allocates at least one more member of one of the three resources that the AAMP has not allocated to that day in its previous iteration.

Proposition 2. *Benders cut (56) is valid Benders feasibility cut.*

Benders optimality cuts. We add optimality cuts to AAMP when $\hat{V}_d^{(i)} < \bar{V}_d^{(i)}$, indicating that sequencing variables and constraints in RSSPs have caused the AAMP-estimated overtime cost $\hat{V}_d^{(i)}$ to increase $\bar{V}_d^{(i)}$.

We develop the following Benders optimality cut:

$$\sum_{m\in\mathcal{M}}\sum_{k\in\hat{\mathcal{K}}_{dm}^{(i)}} C_{mk}\cdot v_{dmk} \geq \bar{V}_d^{(i)}\left(1-\left(\sum_{p\in\hat{\mathcal{P}}_d^{(i)}}(1-w_{pd})+\sum_{m\in\mathcal{M}}\sum_{k\in\left\{\mathcal{K}\setminus\hat{\mathcal{K}}_{dm}^{(i)}\right\}} z_{dmk}\right)\right) \quad \forall d\in\bar{\mathcal{D}}_{\text{sub-optimal}}^{(i)}, \tag{57}$$

where $\bar{\mathcal{D}}_{\text{sub-optimal}}^{(i)}$ is the set of all sub-optimal RSSPs. This cut is an optimality cut because it captures the total overtime cost of each day and enforces AAMP to accept $\bar{V}_d^{(i)}$, change allocation of patients $\hat{\mathcal{P}}_d^{(i)}$, and/or resources in that day $\hat{\mathcal{K}}_{dm}^{(i)}$.

Proposition 3. *Inequality (57) is a valid Benders optimality cut.*

5.5 Implementation

In addition to the choice of partitioning procedure that led to different LBBD variants (Table 4), LBBDs further bifurcate into regular LBBDs (that solve RSSPs when it finds *optimum*) or cutting-plane LBBDs, also known as Branch-and-Checks (B&Cs) (that solve RRSPs for *any AAMP integer feasible solution* and not necessarily optimum). We discuss these two LBBD variants.

5.5.1 LBBD

LBBDs mostly solve their MPs to optimality and provide this optimal solution to SPs (Heching et al., 2019; Hooker, 2007). There are other LBBD variants that allow for the premature stopping of MPs in which case the best integer feasible solution found up to that point is given to SPs (Roshanaei et al., 2017b). The reason for such a premature stopping is the difficulty associated with solving the AAMP and RSSPs that may prevent LBBDs from finding optimal solution within a certain timelimit, hence hindering LBBDs' convergence. Preliminary experiments showed us that the AAMP in LBBD_4 converges to solutions with an average optimality gap of around 2% to 4% within four hours, most of which are used to verify the optimality of solutions of AAMP and RSSPs. Interestingly, the LBBD found comparable results even when we allocated a lower computational time, 3,600 seconds. Specifically, we allocated a maximum of 800 to each iteration of the AAMP. In this case, we use the lower bound of AAMP as the lower bound of the problem. With such time allocation and the consideration of 3600 seconds of runtime, LBBD_4 can iterate at least 6 times on larger instances of the problem. Below, we present LBBD_4 in Algorithm 1.

5.5.2 Cutting-plane LBBDs: Branch-and-Checks

The alternative approach to regular LBBD_4 is to solve RSSPs for each AAMP integer feasible solution using *lazy callbacks* within CPLEX. This approach is known as Branch-and-Check (B&C) (Beck, 2010; Roshanaei et al., 2020a; Thorsteinsson, 2001; Tran et al., 2016), and has been empirically shown to perform well for optimization problems with hard MP and easy SPs (Beck, 2010)—a property that is present in some of our LBBDs. We note that all the optimization components (AAMP, RSSPs, and cuts) are the same in LBBDs and B&Cs and *they only differ when SPs are solved*. Thus, we implement the B&C variants of our LBBDs. In total, we test the performance of four LBBDs and four B&Cs and only present the results associated with the best decomposition algorithm.

Algorithm 1: LBBD$_4$

Input: Use parameters in Table 1
Initialize AAMP, using variables $\mathbf{W}$, $\mathbf{Z}$, and $\mathbf{X}$;
BendersCutPool := $\emptyset$;
while *timelimit is not met* **do**
  Update the MP model with cuts in BendersCutPool;
  Solve the MP to compute the objective, $\hat{Z}_d^{(i)}$ and $\hat{V}_d^{(i)}$, and sets $\hat{\mathcal{P}}_d^{(i)}$, and $\hat{\mathcal{K}}_{dm}^{(i)}$ in Table 6;
  for $d \in \mathcal{D}$ **do**
    Construct the dth RSSP model, using variables $\mathbf{X}$ and $\mathbf{Y}$ for sets $\hat{\mathcal{P}}_d^{(i)}$ and $\hat{\mathcal{K}}_{dm}^{(i)}$;
    Solve the dth SP to compute the objective $\bar{V}_d^{(i)}$;
  end
  Update bounds with $\sum_d \left(\hat{Z}_d^{(i)} + \hat{V}_d^{(i)}\right)$ and $\sum_d \left(\hat{Z}_d^{(i)} + \sum_d \bar{V}_d^{(i)}\right)$;
  if *Optimal (Section 5.4.1)* **then**
    Stop;
  else
    for $d \in \mathcal{D}$ **do**
      if *The dth RSSP is infeasible* **then**
        Develop feasibility Benders cut (56)
      else
        Develop optimality Benders cut (57)
      end
    end
    Add all the developed cuts to the BendersCutPool;
  end
end
Output: bounds

6 Data, parameters, and results

We present our data analysis, computational results, and managerial insights. We first briefly discuss how we prepare our data to find solutions for many clinical questions that are of paramount value for hospital managers. Using this real data, we determine which model or algorithm yields lowest cost within a clinically acceptable timeframe. We elaborate on clinical settings under which each of these algorithms works better. Ascertaining the best algorithm allows us to prescribe our clinical recommendations.

6.1 Dataset

We use a dataset from the General Surgery Department (GSD) of Toronto General Hospital (TGH) from July 2011 to June 2013, consisting of 2711 scheduled surgeries. Emergency surgical patients are handled in a dedicated emergency OR shared among different surgical specialties and hence their scheduling does not impact elective patients. We perform a descriptive analysis on this dataset. We capture the following from this data (i) statistical distribution of surgical durations, (ii) correlation among the parameters of the model (pre-incision, incision, and post-incision times), (iii) possible patients' routes after being discharged from ORs, (iv) performance heterogeneity of surgical resources (nurses, anaesthetists, and surgeons), (v) number of OR-days used and patients scheduled during the studied period, and (vi) the percentage of surgeons changing their ORs within a day (open scheduling) (see Appendix F for complete analysis).

Our data analyses revealed that GSD operates on 25 patients per week, on average (Appendix F.8). As such, we generate a set of benchmark instances, consisting of eight different equal-sized instances of 25

patients. We consider three ORs, five anesthetists, and five surgeons in each instance. Surgical times, time lags between the start and finish of each surgery on each resource, and length of stays in downstream units are directly obtained from the dataset. We determine surgeon eligibility for each surgery in accordance with the set of procedures that each surgeon has historically performed. The cost of ORs and surgeons is obtained from the UHN (University Health Network) case costing department (these estimates have been also used in Roshanaei et al. (2017b)). For anesthetist costs, we assume a uniform distribution between \$600 and \$750 (PayScale, 2018) for each day. We consider 8-PACU capacity per day, 3 ICU beds, and 10-20 Ward beds. We generate 45 instances as follows. We randomly select four surgeons out of 11 surgeons with highest number of performed surgeries. We then randomly select 25 surgeries among those surgeries for which these four surgeons are eligible and available. The reason is to make sure that there is at least one eligible surgeon for each surgery. The average surgeon eligibility for our 45 instances is 2.26 eligible surgeons for each surgery.

We implement MIP models and LBBD algorithms in Python API linked with the IBM ILOG CPLEX 12.8 and IBM ILOGCP Optimizer 12.8. We run experiments on a PC with Core i7 CPU and 16 GB RAM. We consider a computational timelimit equal to 3600 seconds (we also ran the experiments with four hours of computations and obtained almost the same results).

6.2 Which algorithm provides the lowest cost and highest speed in decision-making?

Using our real dataset, (i) we ascertain which solution method (including MIP and CP models as well as LBBD and B&C methods) is best suited for designing TGH's weekly elective surgical schedule, (ii) we reverse engineer final solutions based on which we design and present our effective dominance rules, (iii) we quantify the value of these dominance rules along with the choice of commercial solvers on LBBDs' convergences, and (iv) we study the impact that step-function allocation of overtime to resources (rather than continuous one) yields on the LBBD time and total cost.

Trade-offs for choosing the best algorithm. To determine the best algorithm, we investigate which algorithm finds the best trade-off between *cost-effectiveness* and *decision-making speed*. First, in a budget-constrained hospital like TGH, the number of hours that operating rooms can be used by each specialty is dictated by their allocated budget. Thus, a model that can find more cost-effective solutions allows for scheduling more patients and/or incurring less cost given the same number of patients. Second, operating room scheduling is a challenging task due to dynamic events such as patient cancellation and arrivals of emergency patients all of which require constant rescheduling and decision-making under pressure without the benefit of time and analysis. Therefore, OR managers require a decision support system that can quickly find quality solutions. We aim at improving these two factors, speed and quality of decisions, through our LBBD approaches. The discussion related to why certain decomposition works better than others provides a good theoretical guideline for researchers who wish to develop more effective solution techniques for other difficult operating room scheduling problems.

Initial screening of algorithms' performance. Two of our designed LBBDs, LBBD_1 and LBBD_2, resemble most LBBD algorithms in the literature in that they optimize the entire set of allocation-assignment variables in the AAMP and sequencing variables in their RSSPs. These two LBBDs failed to find a single feasible integer solution for our problem. The performance of LBBD_1 demonstrates the importance of quality SP relaxation for GORPS. With a weak SP relaxation, the solution of the AAMP is likely infeasible because it

over-capacitates RSSPs. The performance of $LBBD_2$ implies the importance of re-optimization in RSSPs. Sequencing RSSPs are mostly infeasible when optimized with unchangeable AAMP-optimized assignment variables. Moreover, Benders cuts that are based on assignment variables are unlikely to change the current AAMP solution due to huge pool of symmetrical solutions. We do not therefore include these two LBBDs in our performance comparison. As stated earlier, this downside (symmetrical solutions) does not exist in the $LBBD_3$ and $LBBD_4$.

Reverse final solution analysis. We discuss how the reverse engineering of final solutions obtained from different instances may help us improve LBBD performances. Having analyzed final solutions, we gathered that the number of daily anesthetists and ORs was the same in the final solution of *all* instances that we solved. Such a finding can partially be ascribed to the cost structure of our resources in that the daily fixed cost associated with using an anesthetist is substantially lower than those of the other two resources: ORs and surgeons. This issue makes the model extremely sensitive to the rather costly use of surgeons' and ORs' overtimes that is penalized with a factor of 1.5, thus precluding the under-staffing of anaesthetists in ORs (under any circumstances) that may delay the start and finish times of these two expensive resources.

Dominance rules. Such a finding (equal number of ORs and anesthetists) allows for the development of two dominance rules that connect the assignment of anesthetists to ORs, alleviating the intractability of the problem significantly. While such a finding immediately reinforces the idea of integrating the cost of ORs with that of anaesthetists, it is not practically possible as anesthetists (like surgeons) may leave ORs before patient removal thus their overtime amounts vary from ORs. Our proposed dominance rules are as follows:

$$z_{d2k} = z_{d1k} \qquad \forall k \in \mathcal{K}_1, d \in \mathcal{D}, \tag{58}$$

$$x_{p2k} = x_{p1k} \qquad \forall p \in \mathcal{P}, k \in \mathcal{K}_1, d \in \mathcal{D}_p. \tag{59}$$

Equation (58) ensures that we assign a *dedicated* anesthetist to each open OR and Equation (59) ensures that if a patient is assigned to an OR, s/he is also assigned to the anesthetist considered for that OR.

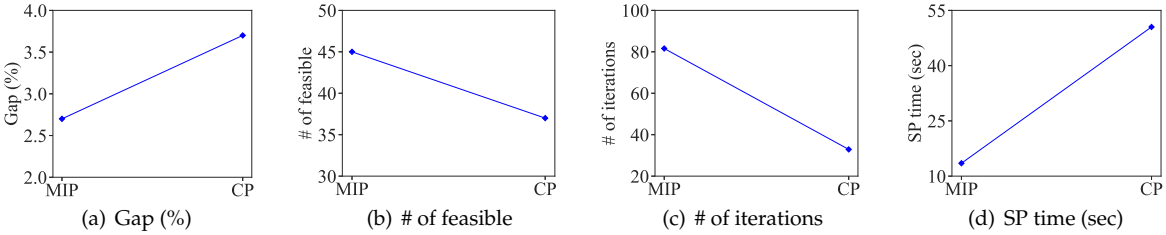
Remark 7. *We state a caveat regarding the inclusion of these dominance rules. Care must be taken with respect to optimality of final solutions as these dominance rules may possibly remove optimal solutions if the availability of anaesthetists and their cost are defined differently than ours. Within the setting of our problem, the final solutions of algorithms with and without these dominance rules are the same.*

Value of dominance rules. We now assess the value of these dominance rules on the tractability of our solution techniques. In Table 7, we compare the performance of these algorithms in terms of optimality gap, number of feasible solutions found, and relative percentage deviation (RPD) of algorithms' integer solutions with respect to the best integer solution in each instance. We show two results from this analysis that dominance rules: (i) yield significant impact on all performance measures when applied to LBBDs; specifically, these dominance rules help $LBBD_3$ and $LBBD_4$ find integer solutions for all instances of the problem, which could not be otherwise achieved (33 and 35 solved instances out of 45, respectively, before the inclusion of these dominance rules) and (ii) yield insignificant impact on optimality gap and feasibility, but slightly improve the quality of integer solutions when incorporated into MIP and CP models (that we measure using RPD).

Table 7: Impact of dominance rules on optimality gaps, RPD, and the number of feasible solutions of solution techniques. "Dom" and "No Dom" represent the problem with and without dominance rules, respectively; average is taken over solved trials; **Bold**: best performance.

Measure	MIP_{SB}		CP		$LBBD_3$		$LBBD_4$	
	Dom	No Dom	Dom	No Dom	Dom	No Dom	Dom	No Dom
Gap (%)	**24.69**	29.02	**70.37**	70.64	**4.11**	6.40	**2.71**	4.09
Feasibility (#)	**45.00**	**45.00**	**45.00**	**45.00**	**45.00**	33.00	**45.00**	35.00
RPD (%)	**2.22**	3.41	**2.41**	2.88	**0.34**	1.22	**0.09**	1.27

LBBD convergence: the impact of CP and MIP for RSSPs. There is a widespread belief in the LBBD literature that solving sequencing RSSPs with CP models (and solving assignment AAMPs with MIP models) accelerate the LBBD convergence. However, we show that our best LBBD ($LBBD_4$) works best when its RSPs are formulated as MIPs and solved via CPLEX rather than CP optimizer (see Figure 6). The choice of MIP over CP for solving SPs allows for lower average SP times, culminating in the $LBBD_4$ being able to iterate more and therefore find more feasible integer solutions with lower average optimality gaps. The primary reason is that the RSSP includes both *assignment* and *sequencing* variables and CP is only good at solving pure *sequencing* problems. On average, CP requires more time to solve RSSPs; as such, the $LBBD_4$ with CP RSSP can complete fewer iterations.



(a) Gap (%) (b) # of feasible (c) # of iterations (d) SP time (sec)

Figure 6: $LBBD_4$ convergence due to the choice of RSSPs

LBBD convergence: breakdown of allocated time and generated cuts. We now analyze the LBBDs' general convergence behavior. In particular, we are interested in scrutinizing how the allotted computational time is allocated to different LBBDs' components (AAMP, RSSPs, and cuts). We show that the proportion of time that each LBBD spends in its AAMP and RSSPs directly impacts the number of iterations (see Table 8). The average AAMP time is increased by 3.6 times (from 38.8 to 142.2 seconds) when we remove dominance rules from the $LBBD_3$. The removal of dominance rules causes the average RSSP time to also double in the $LBBD_3$ due possibly to poor AAMP solutions. The average AAMP time (and almost RSSP time) in $LBBD_4$ remains the same because $LBBD_4$ does not optimize binary decisions for anesthetists in its AAMP. Achieving tractability for the AAMP in $LBBD_4$ (by removing the OR and anesthetist assignment from the AAMP) allows for higher number of iterations. On the positive note, most of the solutions generated by $LBBD_4$ are quickly verified as infeasible by the MIP model, allowing the generation of Benders feasibility cuts that remove these solutions and find new solutions. The average total computational time of $LBBD_4$ with the dominance rules is the lowest since it finds most number of optimal solutions.

Now that we have determined $LBBD_4$ with RSSP-MIP is the best-performing approach and ascertained the underlying reasons for its excellence, we use it to derive our managerial insights.

Table 8: Statistics on the LBBD performance. BFC: Benders feasibility cut; BOC: Benders optimality cut; "Dom" and "No Dom" indicate that the problem has been solved with and without dominance rules, respectively.

Measure	$LBBD_3$		$LBBD_4$	
	Dom	No Dom	Dom	No Dom
Iterations	31.1	11.0	81.6	82.7
BFC	12	7	86	144
BOC	106	36	227	178
Total time (s)	1770	1917	1580	1724
AAMP time (s)	38.8	142.2	5.8	6.2
RSSP time (s)	18.0	32.3	14.5	15.4

LBBD convergence: Impact of step function overtime and resource heterogeneity. We study how computational time (related to convergence) and total cost are affected when hospitals allocate overtime to resources in a step-function manner rather than as continuous one and when the resource performance is heterogeneous versus homogeneous. We solve our instances with $LBBD_4$ when overtime is allocated in increments of 5, 15, and 30 minutes and compare the total cost with that of the continuous overtime (Figure 7). The insight that we acquire from this analysis is that allocation of overtime to resources in a step-function manner yields, unexpectedly, no impact on time and insignificant impact on cost; in fact, the amount of total cost increase due to adopting such an approach remains below 1% (below 0.6% to be more precise) for the values that we considered. As such, our models allow hospital managers to easily adopt such an approach (if legally mandated to do so) without worrying about its time and cost implications.

To compare the impact of capturing performance heterogeneity on the cost function, we compare it with the case that the performances of all members of all resources are homogeneous. The only modification we require to transform heterogeneous performances into homogeneous ones is to take the average of parameter B_{pkm} across all surgeons who can operate on patient p, culminating in using parameter B_{km} that indicates the surgical duration of patient p is independent of the surgeons' performance. It is evident that considering heterogeneity allows us to find more cost-effective assignments for patients from the list of eligible surgeons. The average cost of homogeneous performance is \$86,052, whereas the average heterogeneous cost is \$84,218, i.e., 2.14% higher cost-savings (see Table 9). The insight we acquire from this analysis is that the consideration of heterogeneity in the performance of surgeons improves cost-savings without adverse impact on the complexity of the problem. We can conclude that the complexity of both the heterogeneous and homogeneous problems is almost the same because the average optimality gap of both problems is the same (2.74% vs 3.01%).

Table 9: Impact of heterogeneity on total cost and gap, averaged across 45 instances.

Measure	Heterogeneous			Homogeneous			$\Delta(\%)$		
	Avg.	Max	Min	Avg.	Max	Min	Avg.	Max	Min
Objective (\$)	84,218	113,034	64,477	85,372	114,448	66,024	2.14	6.53	0.11
Gap (%)	2.74	8.21	0	3.01	7.66	0	0.27	3.1	-2.04

6.3 Managerial insights: capturing the performance of the status quo

We address several questions that are of vital importance for our collaborating hospital, TGH, especially during a shock like COVID-19 pandemic. We start this section by quantifying the performance of the status quo based on multiple key performance measures. Note that our analysis of the status quo is based on

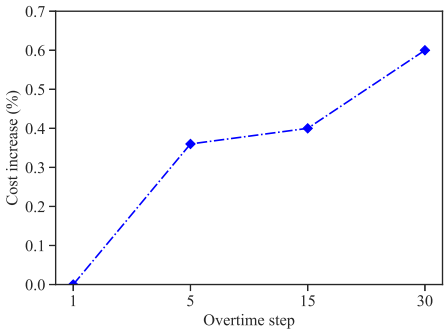


Figure 7: Impact of step-function overtime allocation on total cost.

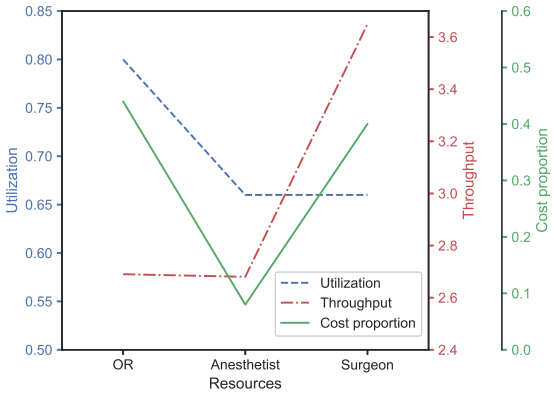


Figure 8: Average resource utilization, throughput, and cost proportion

the average 25 scheduled surgeries per week. We discuss, later in this section, what resource adjustment decisions TGH must make when faced with increased surgical volumes beyond 25 patients per week.

Throughput and utilization analysis. We investigate the quality of final solutions in terms of resource utilization, throughput, and cost proportion (Figure 8). We parse the solutions obtained by $LBBD_4$ on all instances of the problem and report their average values on our key performance indicators. We compute utilization for each resource as the net difference between working hours (e.g., opening hours for ORs) and the actual workload of each member of each resource (e.g., if a surgeon was invited to work for 8 hours and s/he works for 6 hours, her/his utilization is 75%). The utilization of ORs, anesthetists, and surgeons is 80%, 67%, and 67%, respectively. Surgeons incur less wait and idle times than the other two resources as they spend much less time on each surgery and their change of ORs within a day is more cost-effective. Due to effective circumvention of idle and wait times resulted from the adoption of an open scheduling strategy in GORPS, surgeons' throughput are on average 33% higher than those of ORs and anesthetists. In terms of individual cost of each resource, surgeons, ORs, and anesthetists, constitute on average 47%, 43%, and 10% of the total costs, respectively.

Downstream capacities and their impact on feasibility of ORs' schedules. To investigate how downstream capacities (ICU, PACU, and ward beds) impact feasibility of ORs' schedules, we optimize our ORs'

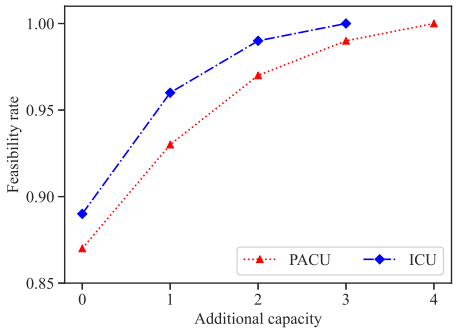


Figure 9: Additional capacity vs feasibility.

schedules without and with downstream capacities. We thus instantiate the constrained model with the optimized solutions from the unconstrained model and capture the number of infeasible instances. We use our 45 instances for this purpose. We discover from the results that only 24% (11 out of 45 instances) of unconstrained solutions are feasible in our constrained optimization problem, demonstrating the importance of considering downstream capacities when optimizing OR schedules. We analyze sources of infeasibility and realize that lack of ICU and PACU beds causes infeasibility in most cases. We also find that ward beds are bottlenecks in only 2% of instances (1 out of 45).

For the reason stated above, we exclude ward beds from our analysis and provide further analyses based on more constraining ICU and PACU capacities.[1] In fact, we are interested in ascertaining how many extra beds of each type we require to ensure that unconstrained OR scheduling optimization yields always-feasible schedules. Unlike the previous analyses that we captured infeasibility in 45 *instances* (each instance can give rise to five daily OR schedules), we now extend our analyses to infeasibility of OR schedules in each *day*. We therefore analyze $45 \times 5 = 225$ daily OR schedules (see Figure 9). This analysis reveals that the optimized unconstrained OR schedules are infeasible in 13% and 11% of days due to limited PACU and ICU beds, respectively. In other words, 87% and 89% of days were feasible despite limited PACU and ICU beds, respectively. The number of feasible OR-days increases to 94% and 100% if we add one and three ICU beds, respectively. Regarding PACU beds, we can ensure feasibility in 96% and 100% of OR-day schedules if we add one and four PACU beds, respectively. Infeasibility in daily OR schedules indicates that *at least* one patient cannot be scheduled in those days. As such, *this analysis helps hospital managers make a trade-off between the lost revenue due to unscheduled patients and the cost of acquiring (and choosing the right selection of) these resources if such a possibility exists.*

6.4 How must TGH react to increase in surgical volumes?

TGH is faced with huge backlog of recently cancelled elective surgeries due to COVID-19 pandemic and also the naturally growing aging population of seniors. Specifically, TGH is interested in finding the best course of actions (capacity acquisition) when increase in surgical volumes is approximately between 20% to 40%. This analysis is quite important for a hospital like TGH that leads Canada in cardiac care, organ transplants, and the treatment of complex patient needs. To evaluate this situation, we generate 30 new

[1]We consider only combinations of number of PACU and ICU beds that were tractable and led to reasonable objective function values. In some cases, increasing the number of beds, beyond what we reported in Figure 9, deteriorated the objective function values due to increased computational complexity.

Table 10: Feasibility and cost trends to increase the capacity of downstream units for patient increase over the 30 larger problems with 30 and 35 patients.

Capacity addition (#)	Feasibility		Cost		Best scenario			Bed added
	#	Ratio	$	Δ	PACU	ICU	Ward	
0	11	37%	102,780	-	0	0	0	-
1	14	47%	102,318	0.45%	0	1	0	ICU
2	18	60%	101,719	1.03%	1	1	0	PACU
3	22	73%	100,749	1.97%	1	1	1	Ward
4	23	77%	100,739	1.99%	2	1	1	PACU
5	26	87%	100,183	2.53%	2	1	2	ICU
6	27	90%	100,161	2.55%	2	2	2	Ward

larger instances. Precisely, we extend the size of instances from 25 patients to 30 (20 instances) and to 35 (10 instances) patients per week. Initial analyses revealed that the current number of ORs (and the possible use of overtime) suffice to accommodate higher number of patients. We thus provide recommendations solely on how downstream capacities (ICU, PACU, and ward beds) must be adjusted to accommodate the possible increase in surgical volumes. We study feasibility and cost implications of surgical volume and capacity increases (Table 10).

Downstream capacity adjustments due to surgical volume increase. We first solve our new instances with the same downstream capacities and gradually increase capacities (one at a time) to capture their impacts on feasibility and cost of OR schedules. We discuss, among downstream capacities, which one is more constraining and thus needs its capacity to be increased first (see Table 10 for details). With existing level of downstream capacities, only 37% of new instances are feasible, demonstrating the need for downstream capacity expansion. The first question that naturally arises in such a situation is that "how many downstream beds and of which type must TGH add to better handle the increase in its surgical volume?" Our results showed that increasing an ICU bed increases feasibility from 37% to 47% and decreases cost (related to ORs and resources therein) by 0.45%. Increasing feasibility by 10% corresponds to scheduling one to two more patients per week, on average. According to Roshanaei et al. (2017b), the average benefit of scheduling one more patient in TGH (and the University Health Network that encompasses TGH, in general) is 2,600CAD (see Section 3.2). While we do not have estimates for the cost of adding each ICU bed (and downstream beds, in general), the hospital manager can trade off the potential long-term revenue stream due to increased weekly number of surgeries (and the already reduced cost mentioned in Table 10) with the one-time cost of adding an ICU bed. We show that by adding two beds in each downstream unit of ORs, TGH can achieve 90% feasibility (i.e., TGH is able to manage the increased weekly surgical demand 90% of times) and 2.55% cost savings. We provide a visualization of these results in Figure 10.

Cost and revenue implications of surgical volume increase. We discussed implications of surgical volume increase on downstream requirements. We now provide some statistics regarding *OR scheduling cost increase* due to surgical volume increase given that we consider two more units of capacity for each downstream resource. (see Table 11) We provide cost averages (among other information) for our 75 instances, consisting of 45, 20, and 10 instances of patient sizes 25, 30, and 35, respectively (Table 11). As we increase the size of surgical volume, the average optimality gap of larger instances expectedly increases, from 2.7% to 6.5%. While we linearly vary surgical volumes from 25 to 35 with an increment of five patients, 20% increase with respect to base patient population (see Figure 11), the total cost increase due to surgical vol-

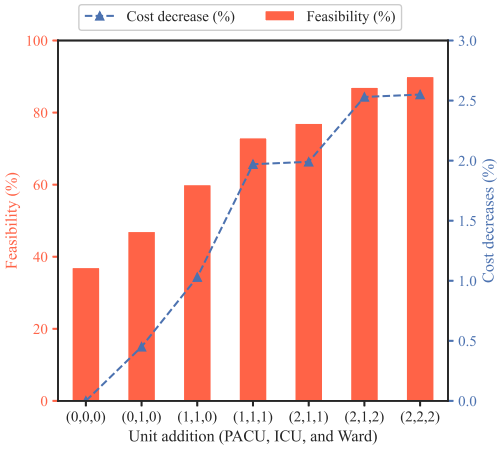


Figure 10: Capacity addition for patient increase: feasibility and cost

Table 11: LBBD performance

Size	#	Cost ($)	Gap (%)	Itr (#)	MP time (s)	SP time (s)	Δ Cost	Δ Itr	Δ MP time	Δ SP time
25	45	84,218	2.7	82	6	13	-	-	-	-
30	19	99,120	5.4	38	14	41	18%	-53%	147%	205%
35	8	107,996	6.5	21	14	66	29%	-74%	145%	136%

ume increase grows with a lower rate of 18% (for 20% increase) and 29% (for 40% increase). Based on this analysis, we recommend that TGH schedule 35 surgeries rather than 30 if two additional downstream beds are added to each of its downstream units. Having conducted this analysis, we realized that the per-patient OR scheduling cost is reduced by CAD$283 that corresponds to 8.4% cost-savings compared to when TGH schedules only 25 patients, on average. This cost-saving is in addition to the average revenue that each of the new 10 patients will generate for the hospital (CAD$2,600 per patient), i.e., additional $1,468,000 annual revenue.

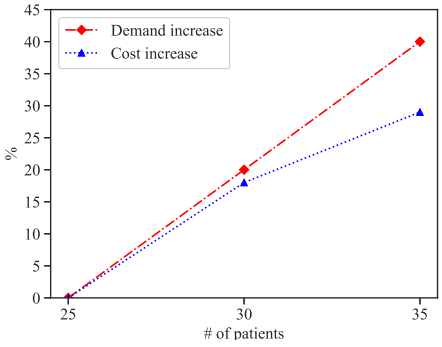


Figure 11: Demand increase vs cost increase

7 Concluding remarks

Problem definition and general takeaways: We studied an operating planning and scheduling problem in the Toronto General Hospital (TGH), Ontario, Canada. Inspired by the literature and scheduling needs of TGH, we developed for them a model that captures the scheduling of all critical resources in an OR and their dynamics in terms of lead and lag times between their start times on each surgery, their eligibility for each surgery, and also their performance heterogeneity, among many other relevant operational factors. We called this problem *generalized* operating room planning and scheduling (GORPS) because our models help the nurse manager in TGH determine the right *number*, *selection*, and *overtime* of resources for the set of patients scheduled on each day. Furthermore, our models allow the nurse manager to determine the optimal level of overtime allocation for each member of each resource in each day. We showed how our models contribute to the theory and practice of operating room scheduling by capturing many relevant operational factors (that are considered in both private and public hospitals). We developed an exact decomposition design that works well in particularly with heterogeneous resources which are not only in healthcare but also in many other industries such as such as transportation, supply chain and manufacturing. We obtained important and practical insights which are useful not only for TGH but also for other similar hospitals.

Models, methods, and algorithmic insights. We developed various deterministic models to improve *solution quality* and *decision-making speed* for GORPS. In fact, we conducted the first comprehensive evaluation of mathematical and constraint programming models thanks to the new advances in CP optimizer that solve CP models, we can capture solution quality of each integer solution found and learned that none of these models are efficient enough for solving GORPS in a real practical setting due to its complexity. We thus exploited the structural decomposability of our models and developed efficient decomposition techniques based on logic-based Benders decomposition (LBBD) algorithms and their cutting-plane-based counterparts, Branch-and-Check (B&C). We showed three counter-intuitive algorithmic results. First, our best mathematical programming model achieves an average optimality gap of 24.69%, whereas the CP model obtains an average optimality gap of 70.37%. We additionally showed that CP is not even effective for solving the sequencing subproblems (which are much smaller in size and have much fewer variables) in our LBBD—a capability in CP that has been commended in all CP-based LBBD algorithms that hybridize integer programming with constraint programming. This finding flies in the face of many recent studies that show the superiority of CP models for solving general scheduling problems. Second, our best LBBD achieves an average optimality gap of 2.71% and that our B&C that solves subproblems for each integer solution (rather than the optimal solution) of the master problem is not an efficient approach for solving GORPS. Third, unlike the literature that seeks all possible ways to be able to solve the sequencing subproblems (and subproblems, in general) in a polynominal time, we can achieve better results if sequencing SPs are more difficult to solve, but can provide more accurate information (tighter solutions) to the master problem. In fact, we assessed the performance of two LBBDs, (i) one with most information (variables and constraints) centralized in the master problem and least information in the subproblems and (ii) the other one with balanced information in the master problem and sub-problems. The latter worked better even when its subproblems were more difficult to solve. We concluded the algorithmic section by providing an insight that the way overtime is allocated to resources (continuous versus step-function) does not worsen complexity of models and methods, but insignificantly increases the total cost.

Data analysis and managerial insights. In addition to information that we gathered from the nurse manager and Dr. David R. Urbach (who is the surgeon-in-chief and medical director of the Women's College Hospital and Senior Scientist at Toronto General Research Institute) at TGH during our interviews, we conducted an

in-depth data analysis to identify missing information so as to develop more likely implementable models that capture most of OR scheduling needs at TGH. After applying our efficient decomposition techniques to the real data, we analyzed TGH's OR scheduling performance from various perspectives and provided a variety of managerial insights. We started our analyses by assessing the impact of downstream capacities on the feasibility of OR schedules. We showed that OR schedules optimized in the absence of downstream capacities are feasible only in 24% of times when examined against downstream capacities. We showed that in most cases lack of PACU and ICU beds were primary and secondary sources of infeasibility followed by the ward beds as the tertiary source of infeasibility. Our $LBBD_4$ (as the most efficient algorithm for solving GORPS) finds solutions that significantly increase utilization of all resources working in an operating room, and surgeons as the most expensive resource, in particular. We analyzed different scenarios and proposed effective strategies for downstream capacity adjustments when TGH faced with surgical volume increase such as the current situation due to Covid-19 pandemic. We found that operating rooms were not a constraining resource and could accommodate much higher number of surgeries should TGH prudently increase its downstream capacities. Significant revenue could be generated due to (i) increased number of surgeries and (ii) achieved ORs' economies of scale that directly impact per-patient surgery cost-saving. This conceivable revenue could be attained if TGH invested in its downstream capacities. Last but not least, our study is not without any limitations. Our models are deterministic although some of the parameters for our problem setting can be stochastic. We conducted analysis on the uncertainty of pre-incision, incision, and post-incision times; however, the computational complexity of the problem prevented us from obtaining any practical generalizable insights. Therefore, we chose the deterministic modelling approach in which we were able to develop more efficient solution methods and carry out more experiments to acquire new and more insights for generalized operating room planning and scheduling problem.

Avenues for future research. For future algorithmic work, a simulation-optimization approach used in Chow et al. (2011) and heuristic algorithms developed in Gul et al. (2011) could be considered for the stochastic variant of GORPS. It would to be interesting to examine the performance of LBBDs under parallel sub-problem computation. Other cut strengthening strategies similar to the ones proposed in Bodur et al. (2017); Bodur and Luedtke (2017); Hooker (2007) can be examined for the deterministic and stochastic variants of GORPS. Finally, the sequencing sub-problems could be solved by multivalued decision diagrams (Cire and van Hoeve, 2013). GORPS models a single surgical specialty (e.g., General Surgery Department in this paper). The mathematical models that we developed for GORPS can exactly be used by other surgical specialties because, in the master surgical scheduling phase, resources (ORs and downstream beds) are optimally allocated to each specialty (resources are decomposed based on specialties), leading to independent surgery scheduling process for each specialty (in most cases when no off-serving is allowed in hospitals). The GORPS mathematical structure allows the handling of other variations wherein surgical equipment and surgeon assistants are shared among ORs. The lag time, similar to the one considered for surgeons after anesthesia, should be also considered for surgical equipment shared among ORs, because they need to be sterilized after each use. Batun et al. (2011) showed that surgeons need to only stay in ORs for the critical portion of a surgery since other non-critical portions can be completed by surgeon assistant(s). GORPS can be slightly modified to encompass situations where surgeon assistants are able to perform the first and/or the last portion of a surgery within an OR. GORPS requires a significant change if the same surgeon assistant is concurrently assigned to the first and last portion of a surgery, but s/he could potentially join other ORs while the critical portion of his/her previously assigned surgery is being performed by the surgeon. This variation is extremely computationally challenging, because it requires numerous sequencing constraints to ensure the feasibility of the schedule.

Appendix

We include the mixed-integer models (with continuous overtime and step function overtime), proofs, and data analysis in this appendix.

A Mixed-integer models with continuous overtime

A.1 Immediate-sequence-based MIP (MIP_{ISB}) paradigm

Similar to MIP_{SB}, MIP_{ISB} models investigate sequencing between any pair of surgeries that can be assigned to a member of a resource in each day, but the precedence between any two arbitrary surgeries is considered immediate. The immediacy of sequencing each pair of surgeries is shown in binary sequencing variable $y_{pdmkp'} \mid p \neq p'$ (see Table 12 for the definition). The use of the new sequencing variable in MIPISB leads to a complete transformation of the model with various number of variables and constraints. We use the notion of a 0th surgery to formulate the GORPS model under this paradigm. MIP_{ISB} model is as follows:

Table 12: Sequencing variable used in immediate sequence-based modeling paradigm

Sequencing	$y_{pdmkp'} \in \{0,1\}$	1 if patient p is operated on immediately after patient p' ($p \neq p'$) by kth member of resource m on day d, 0 otherwise

$$\text{minimize} \quad \sum_{m\in\mathcal{M}} \sum_{k\in\mathcal{K}_m} \sum_{d\in\mathcal{D}_{mk}} \left(F_{mk} z_{dmk} + C_{mk} v_{dmk} \right) \tag{MIP$_{\text{ISB}}$}$$

subject to Constraints (1), (9) - (18)

$$\sum_{k\in\mathcal{K}_{mp}} \sum_{p'\in\{0\,\cup\,\mathcal{P}\}|p'\neq p, k\in\mathcal{K}_{mp'}} \sum_{d\in\mathcal{D}_p\cap\mathcal{D}_{p'}\cap\mathcal{D}_{mk}} y_{pdmkp'} = 1 \qquad \forall p\in\mathcal{P}, m\in\mathcal{M} \tag{60}$$

$$\sum_{p\in\mathcal{P}|p'\neq p, k\in\mathcal{K}_{mp}} \sum_{d\in\mathcal{D}_p\cap\mathcal{D}_{p'}\cap\mathcal{D}_{mk}} y_{pdmkp'} \leq 1 \qquad \forall p^{'} > 0, m\in\mathcal{M}, k\in\mathcal{K}_{mp'} \tag{61}$$

$$y_{pdmkp'} \leq w_{pd}$$
$$\forall p\in\mathcal{P}, p'\in\{0\,\cup\,\mathcal{P}\}|p'\neq p, m\in\mathcal{M}, k\in\mathcal{K}_{mp}\cap\mathcal{K}_{mp'}, d\in\mathcal{D}_p\cap\mathcal{D}_{p'}\cap\mathcal{D}_{mk} \tag{62}$$

$$y_{pdmkp'} \leq z_{dmk}$$
$$\forall p\in\mathcal{P}, p'\in\{0\,\cup\,\mathcal{P}\}|p'\neq p, m\in\mathcal{M}, k\in\mathcal{K}_{mp}\cap\mathcal{K}_{mp'}, d\in\mathcal{D}_p\cap\mathcal{D}_{p'}\cap\mathcal{D}_{mk} \tag{63}$$

$$\sum_{p\in\mathcal{P}|k\in\mathcal{K}_{mp}, d\in\mathcal{D}_p} y_{pdmk0} = z_{dmk} \qquad \forall m\in\mathcal{M}, k\in\mathcal{K}_m, d\in\mathcal{D}_{mk} \tag{64}$$

$$\sum_{p'\in\mathcal{P}|p'\neq p, k\in\mathcal{K}_{mp'}, d\in\mathcal{D}_p} y_{p'dmkp} \leq \sum_{p'\in\{0\,\cup\,\mathcal{P}\}|p'\neq p, k\in\mathcal{K}_{mp'}, d\in\mathcal{D}_{p'}} y_{pdmkp'}$$
$$\forall p\in\mathcal{P}, m\in\mathcal{M}, k\in\mathcal{K}_{mp}, d\in\mathcal{D}_{mk} \tag{65}$$

$$c_{pm} \geq s_{pm} + \sum_{p'\in\{0\,\cup\,\mathcal{P}\}|p'\neq p} \sum_{k\in\{\mathcal{K}_{mp}\cap\mathcal{K}_{mp'}\}} \sum_{d\in\mathcal{D}_p\cap\mathcal{D}_{p'}\cap\mathcal{D}_{mk}} B_{pmk} y_{pdmkp'} \qquad \forall p\in\mathcal{P}; m\in\mathcal{M} \tag{66}$$

$$s_{pm} \geq c_{p'm} - M_3 \left(1 - \sum_{d\in\mathcal{D}_p\cap\mathcal{D}_{p'}\cap\mathcal{D}_{mk}} y_{pdmkp'} \right)$$

Table 13: Notation for position-based modeling paradigm

Sets:	
$\mathcal{L}$	Set of priority positions, $\ell \in \mathcal{L}$, $\lvert\mathcal{L}\rvert = \lvert\mathcal{P}\rvert$
Variables:	
$x_{\ell mk}$	1 if patient in position ℓ is assigned to kth member of resource m, 0 otherwise
$y_{p\ell d}$	1 if patient p is operated on day d in position ℓ, 0 otherwise
$w_{\ell d}$	1 the patient in position ℓ is assigned to day d, 0 otherwise
z_{dmk}	1 if kth member of resource m is used on day d, 0 otherwise
$s_{\ell m}$	Starting time of resource m on the patient in position ℓ
$c_{\ell m}$	Completion time of resource m on the patient in position ℓ
$f_{\ell dmk}$	Finishing time of kth member of resource m for the patient in position ℓ on day d
v_{dmk}	Overtime amount of kth member of resource m on day d

$$\forall p, p' \in \mathcal{P} | p \neq p', m \in \mathcal{M}, k \in \mathcal{K}_{mp} \cap \mathcal{K}_{mp'} \tag{67}$$

$$f_{pdmk} \geq c_{pm} - M_3\left(1 - \sum_{p' \in \{0 \cup \mathcal{P}\} | p' \neq p, k \in \mathcal{K}_{mp'}, d \in \mathcal{D}_{p'}} y_{pdmkp'}\right)$$

$$\forall p \in \mathcal{P}, m \in \mathcal{M}, k \in \mathcal{K}_{mp}, d \in \mathcal{D}_p \cap \mathcal{D}_{mk} \tag{68}$$

$$y_{pdmkp'}, z_{dmk}, w_{pd} \in \{0,1\} \qquad \forall p \in \mathcal{P}, p' \in \{0 \cup \mathcal{P}\} | p' \neq p, d \in \mathcal{D}, m \in \mathcal{M}, k \in \mathcal{K}_{mp} \tag{69}$$

Constraint (60) determines the proceeding of each patient at the sequence for each resource. Constraint (61) ensures each patient having at most one successor in the sequence. Constraint (62) allows each patient to appear in the sequence on the day assigned to. Constraint (63) specifies the members of resources used on each day. Constraint (64) ensures one sequence for each member of resource on the day they are used. Constraint (65) limits the precedence relationship only for surgeries assigned to the same member of each resource in each day. Constraints (66), (66), (67) build a feasible schedule similar to constraints (5), (6), (7), and (8) in MIP_{SB}. Constraint (69) is to define the binary variables.

A.2 Position-based MIP (MIP_{PB}) paradigm

MIP_{PB} paradigm is an assignment problem where each patient should be assigned to exactly one position within the list of a member of a resource, but each position can be simultaneously occupied by more than one member of each resource. We note that the cardinality of the set of positions is equal to that of the set of patients in MIP_{PB} ($|\mathcal{L}| = |\mathcal{P}|$), ensuring that each position is exactly assigned to a patient. While these two sets can be used interchangeably, we differentiate them for clarity. Surgery sequencing based on position-based modeling requires defining a new binary variable ($y_{p\ell}$) to assign each patient to a position within the list of a member of a resource. Notation is shown in Table 14. The MIP_{PB} is as follows:

$$\text{minimize} \quad \sum_{m \in \mathcal{M}} \sum_{k \in \mathcal{K}_m} \sum_{d \in \mathcal{D}_{mk}} \left(F_{mk} z_{dmk} + C_{mk} v_{dmk}\right) \tag{MIP_{PB}}$$

$$\text{subject to} \quad \text{Constraints (12), (58), and (59)}$$

$$\sum_{d \in \mathcal{D}} w_{\ell d} = 1 \qquad \forall \ell \in \mathcal{L} \tag{70}$$

$$\sum_{p\in\mathcal{P}|d\in\mathcal{D}_p} y_{p\ell d} = w_{\ell d} \qquad \forall \ell \in \mathcal{L}, d \in \mathcal{D} \tag{71}$$

$$\sum_{\ell\in\mathcal{L}}\sum_{d\in\mathcal{D}_p} y_{p\ell d} = 1 \qquad \forall p \in \mathcal{P} \tag{72}$$

$$\sum_{k\in\mathcal{K}_m} x_{\ell mk} = 1 \qquad \forall \ell \in \mathcal{L}, m \in \mathcal{M} \tag{73}$$

$$x_{\ell mk} \leq \sum_{d\in\mathcal{D}_{mk}} w_{\ell d} \qquad \forall \ell \in \mathcal{L}, m \in \mathcal{M}, k \in \mathcal{K}_m \tag{74}$$

$$x_{\ell mk} \leq 1 - y_{p\ell d} \quad \forall \ell \in \mathcal{L}, p \in \mathcal{P}, m \in \mathcal{M}, k \in \mathcal{K}_m \setminus \mathcal{K}_{mp}, d \in \mathcal{D}_p \cap \mathcal{D}_{mk} \tag{75}$$

$$x_{\ell mk} + w_{\ell d} \leq 1 + z_{dmk} \qquad \forall \ell \in \mathcal{L}, m \in \mathcal{M}, k \in \mathcal{K}_m, d \in \mathcal{D}_{mk} \tag{76}$$

$$c_{\ell m} \geq s_{\ell m} + B_{pmk}\big(y_{p\ell d} + x_{lmk} - 1\big) \quad \forall \ell \in \mathcal{L}, p \in \mathcal{P}, m \in \mathcal{M}, k \in \mathcal{K}_{mp}, d \in \mathcal{D}_p \cap \mathcal{D}_{mk} \tag{77}$$

$$s_{\ell m} \geq c_{\ell' m} - M(4 - x_{\ell mk} - x_{\ell' mk} - w_{\ell d} - w_{\ell' d}) \qquad \forall (\ell, \ell') \in \mathcal{L} \mid \ell > \ell',\ m \in \mathcal{M}, k \in \mathcal{K}_m, d \in \mathcal{D}_{mk} \tag{78}$$

$$f_{\ell dmk} \geq c_{\ell m} - M(2 - x_{\ell mk} - w_{\ell d})\ \forall \ell \in \mathcal{L}, m \in \mathcal{M}, k \in \mathcal{K}_m, d \in \mathcal{D}_{mk} \tag{79}$$

$$s_{\ell m} \geq s_{\ell,m-1} + \sum_{p\in\mathcal{P}}\sum_{d\in\mathcal{D}_p} G_{pm} \cdot y_{p\ell d} \forall \ell \in \mathcal{L}, m \in \mathcal{M} \setminus 1 \tag{80}$$

$$c_{\ell 1} \geq c_{\ell 3} + \sum_{p\in\mathcal{P}}\sum_{d\in\mathcal{D}_p} E_p \cdot y_{p\ell d} \qquad \forall \ell \in \mathcal{L} \tag{81}$$

$$c_{p3} \leq c_{p2} + \sum_{p\in\mathcal{P}}\sum_{d\in\mathcal{D}_p} W_p \cdot y_{p\ell d} \qquad \forall \ell \in \mathcal{L} \tag{82}$$

$$\sum_{p\in\mathcal{P}_1|d\in\mathcal{D}_p}\sum_{\ell\in\mathcal{L}} y_{p\ell d} \leq \alpha \qquad \forall d \in \mathcal{D} \tag{83}$$

$$\sum_{p\in\mathcal{P}_2|d\in\mathcal{D}_p}\sum_{\ell\in\mathcal{L}}\sum_{d'=\max\{1,d-H_p+1\}}^{d} y_{p\ell d'} \leq \beta_d \quad \forall d \in \mathcal{D} \tag{84}$$

$$\sum_{p\in\{\mathcal{P}_3\setminus\mathcal{P}_2\}|d\in\mathcal{D}_p}\sum_{\ell\in\mathcal{L}}\sum_{d'=\max\{1,d-A_p+1\}}^{d} y_{p\ell d'} + \sum_{p\in\{\mathcal{P}_3\cap\mathcal{P}_2\}|d-H_p\geq 1,d\in\mathcal{D}_P}\sum_{\ell\in\mathcal{L}}\sum_{d'=\max\{1,d-H_p-A_p+1\}}^{d-H_p} y_{pd\ell'} \leq \gamma_d \quad \forall d \in \mathcal{D} \tag{85}$$

$$w_{\ell d}, x_{\ell mk}, z_{dmk}, y_{p\ell d} \in \{0,1\} \quad \forall \ell \in \mathcal{L}, p \in \mathcal{P}, d \in \mathcal{D}, m \in \mathcal{M}, k \in \mathcal{K}_m \tag{86}$$

Constraint (70) assigns each position to exactly one day in a week. Constraint (71) assigns each surgery to exactly one of the existing positions, while Constraint (72) assigns each position to one surgery. Constraint (73) assigns one resource to each surgery position. Constraint (74) ensures the day availability of resources. Constraint (75) removes ineligible assignment of resources to patients. Constraint (76) determines the resources used. Constraint (77) makes sure that the difference between the starting and completion time of each surgery in each position for each member of each resource is at least as large as its processing time. Constraint (78) ensures that no two positions of different members of different resources on each day overlap. Constraint (79) ensures that the finishing time of each position assigned to each member of each resource on each day is at least as large as the completion time of the last position assigned to that member on that day. Constraints (80), (81), and (82) ensure that the time lags are met. Constraints (83), (84), (85) ensures the availability of downstream units. The remaining Constraint (86) is for the corresponding binary

Table 14: Notation for time-based modeling paradigm

Sets:	
$\mathcal{T}_{mk}$	Set of time units for kth member of resource m, $t \in \mathcal{T}_{mk}$, $\lvert\mathcal{T}_{mk}\rvert = T + V_{mk}$
Variables:	
y_{pdtmk}	1 if patient p is operated on day d at time t by kth member of resource m, 0 otherwise
x_{pmk}	1 if patient p is operated by by kth member of resource m, 0 otherwise
w_{pd}	1 the patient p is assigned to day d, 0 otherwise
z_{dmk}	1 if kth member of resource m is used on day d, 0 otherwise
s_{pm}	Starting time of resource m on the patient p
c_{pm}	Completion time of resource m on the patient p
v_{dmk}	Overtime amount of kth member of resource m on day d

variables.

A.3 Time-based MIP (MIP$_{\text{TB}}$) paradigm

MIP$_{\text{TB}}$ paradigm is likely the most popular paradigm among OR scheduling researchers (Hashemi Doulabi et al., 2016; Marques et al., 2012; Vijayakumar et al., 2013). Notation is given in Table 14. MIP$_{\text{TB}}$ discretizes time parameters (e.g., B_{pmk}) into arbitrary time slots of equal sizes, say, 1, 5, or 15 minutes and ensures that the total time slots assigned to each surgery is equal to the time that has been considered for that surgery. The time-based MIP, denoted MIP$_{\text{TB}}$, is as follows:

$$\text{minimize} \quad \sum_{d\in\mathcal{D}}\sum_{m\in\mathcal{M}}\sum_{k\in\mathcal{K}_m}\Big(F_{mk}z_{dmk} + C_{mk}v_{dmk}\Big) \tag{MIP$_{\text{TB}}$}$$

subject to Constraints (1), (2), (4), (9), (10), (11), (13), (14), (15), (58), and (59)

$$\sum_{d\in\mathcal{D}_p\cap D_{mk}}\sum_{t\in\mathcal{T}_{mk}} y_{pdtmk} = B_{pmk}\cdot x_{pmk} \qquad \forall p\in\mathcal{P}, m\in\mathcal{M}\setminus 1, k\in\mathcal{K}_{mp} \tag{87}$$

$$\sum_{d\in\mathcal{D}_p\cap D_{1k}}\sum_{t\in\mathcal{T}_{1k}} y_{pdt1k} \geq \sum_{k'\in\mathcal{K}_{1p}} B'_{p1k}\cdot x_{p3k'} - M(1-x_{p1k}) \qquad \forall p\in\mathcal{P}, k\in\mathcal{K}_{mp} \tag{88}$$

$$x_{pdtmk} \leq w_{pd} \qquad \forall p\in\mathcal{P}, t\in\mathcal{T}_{mk}, m\in\mathcal{M}, k\in\mathcal{K}_{mp}, d\in\mathcal{D}_p\cap D_{mk} \tag{89}$$

$$y_{pdtmk} \leq z_{pmk} \qquad \forall p\in\mathcal{P}, t\in\mathcal{T}_{mk}, m\in\mathcal{M}, k\in\mathcal{K}_{mp}, d\in\mathcal{D}_p\cap D_{mk} \tag{90}$$

$$\sum_{p\in\mathcal{P}|k\in\mathcal{K}_{mp}, d\in\mathcal{D}+p} y_{pdtmk} \leq 1 \qquad \forall t\in\mathcal{T}_{mk}, m\in\mathcal{M}, k\in\mathcal{K}_m, d\in D_{mk} \tag{91}$$

$$s_{pm} \leq t\cdot y_{pdtmk} + |\mathcal{T}_{mk}|(1-y_{pdtmk}) \qquad \forall p\in\mathcal{P}, t\in\mathcal{T}_{mk}, m\in\mathcal{M}, k\in\mathcal{K}_{mp}, d\in\mathcal{D}_p\cap D_{mk} \tag{92}$$

$$c_{pm} \geq t\cdot y_{pdtmk} \qquad \forall p\in\mathcal{P}, t\in\mathcal{T}_{mk}, m\in\mathcal{M}, k\in\mathcal{K}_{mp}, d\in\mathcal{D}_p\cap D_{mk} \tag{93}$$

$$c_{pm} - s_{pm} \leq \sum_{k\in\mathcal{K}_{mp}} (B_{pmk}-1)\cdot x_{pmk} \quad \forall p\in\mathcal{P}, m\in\mathcal{M}\setminus 1 \tag{94}$$

$$c_{pm} - s_{pm} \leq \sum_{k\in\mathcal{K}_{Mp}} (B'_{pmk}-1)\cdot y_{pMk} \quad \forall p\in\mathcal{P}, m=1 \tag{95}$$

$$v_{dmk} \geq c_{pm} - V(2-x_{pmk}-w_{pd}) \quad \forall p\in\mathcal{P}, m\in\mathcal{M}, k\in\mathcal{K}_{mp}, d\in\mathcal{D}_p\cap D_{mk} \tag{96}$$

$$s_{pm}, c_{pm}, v_{dmk} \geq 0 \qquad \forall p\in\mathcal{P}, d\in\mathcal{D}, m\in\mathcal{M}, k\in\mathcal{K}_{mp} \tag{97}$$

$$w_{pd}, y_{pdtmk}, z_{dmk}, x_{pmk} \in \{0,1\} \qquad \forall p\in\mathcal{P}, d\in\mathcal{D}, t\in\mathcal{T}_{mk}, m\in\mathcal{M}, k\in\mathcal{K}_m \tag{98}$$

Table 15: Parametric size dimensionality of MIPs; $|\mathcal{P}|, |\mathcal{M}|$, and $|\mathcal{K}|$ represent the number of patients, resources, and members for each resource, respectively. All the models require the same number of binary variables for allocation ($\boldsymbol{W}$, $\boldsymbol{Z}$) and assignment ($\boldsymbol{X}$), which is $|\mathcal{P}| \cdot |\mathcal{D}| + |\mathcal{D}| \cdot |\mathcal{M}| \cdot |\mathcal{K}|$ and $|\mathcal{P}| \cdot |\mathcal{M}| \cdot |\mathcal{K}|$, respectively.

Model	Feature	#
MIP_{SB}	Binary variables (Y)	$\frac{|\mathcal{P}|(|\mathcal{P}|-1)}{2}$
	Continuous variables	$2\|\mathcal{P}\| \cdot \|\mathcal{M}\| + \|\mathcal{P}\| \cdot \|\mathcal{D}\| \cdot \|\mathcal{M}\| \cdot \|\mathcal{K}\| + \|\mathcal{D}\| \cdot \|\mathcal{M}\| \cdot \|\mathcal{K}\|$
	Constraints	$3\|\mathcal{P}\| + 2\|\mathcal{P}\| \cdot \|\mathcal{M}\| + \|\mathcal{P}\| \cdot \|\mathcal{M}\| \cdot \|\mathcal{K}\|(3\|\mathcal{D}\| + 1) + \|\mathcal{P}\| \cdot \|\mathcal{K}\| + \|\mathcal{P}\|^2 \cdot \|\mathcal{M}\| \cdot \|\mathcal{K}\| \cdot \|\mathcal{D}\| + 3\|\mathcal{D}\|$
MIP_{ISB}	Binary variables (Y)	$\|\mathcal{P}\|^2 \cdot \|\mathcal{D}\| \cdot \|\mathcal{M}\| \cdot \|\mathcal{K}\|$
	Continuous variables	$2\|\mathcal{P}\| \cdot \|\mathcal{M}\| + \|\mathcal{P}\| \cdot \|\mathcal{D}\| \cdot \|\mathcal{M}\| \cdot \|\mathcal{K}\| + \|\mathcal{D}\| \cdot \|\mathcal{M}\| \cdot \|\mathcal{K}\|$
	Constraints	$3\|\mathcal{P}\| + 3\|\mathcal{P}\| \cdot \|\mathcal{M}\| + \|\mathcal{P}\| \cdot \|\mathcal{M}\| \cdot \|\mathcal{K}\|(3\|\mathcal{D}\| + 1) + 3\|\mathcal{D}\| + \|\mathcal{P}\|^2 \cdot \|\mathcal{M}\| \cdot \|\mathcal{K}\|(2\|\mathcal{D}\| + 1) + \|\mathcal{M}\| \cdot \|\mathcal{K}\| \cdot \|\mathcal{D}\|$
MIP_{PB}	Binary variables (Y)	$\|\mathcal{P}\|^2 \cdot \|\mathcal{D}\|$
	Continuous variables	$2\|\mathcal{P}\| \cdot \|\mathcal{M}\| + \|\mathcal{P}\| \cdot \|\mathcal{D}\| \cdot \|\mathcal{M}\| \cdot \|\mathcal{K}\| + \|\mathcal{D}\| \cdot \|\mathcal{M}\| \cdot \|\mathcal{K}\|$
	Constraints	$\|\mathcal{P}\| \cdot \|\mathcal{M}\| \cdot \|\mathcal{K}\|(3\|\mathcal{D}\| + 1) + 4\|\mathcal{P}\| + \|\mathcal{P}\| \cdot \|\mathcal{D}\| + 2\|\mathcal{P}\| \cdot \|\mathcal{M}\| + 1.5\|\mathcal{P}\|^2 \cdot \|\mathcal{M}\| \cdot \|\mathcal{K}\| \cdot \|\mathcal{D}\| + 3\|\mathcal{D}\|$
MIP_{TB}	Binary variables (Y)	$\|\mathcal{P}\| \cdot \|\mathcal{D}\| \cdot (T + V) \cdot \|\mathcal{M}\| \cdot \|\mathcal{K}\|$
	Continuous variables	$2\|\mathcal{P}\| \cdot \|\mathcal{M}\| + \|\mathcal{D}\| \cdot \|\mathcal{M}\| \cdot \|\mathcal{K}\|$
	Constraints	$4\|\mathcal{P}\| + 3\|\mathcal{P}\| \cdot \|\mathcal{M}\| + 2\|\mathcal{P}\| \cdot \|\mathcal{K}\| + \|\mathcal{P}\| \cdot \|\mathcal{M}\| \cdot \|\mathcal{K}\|(\|\mathcal{D}\| + 1) + 3\|\mathcal{D}\| + (\|T\| + \|V\|)\|\mathcal{M}\| \cdot \|\mathcal{K}\| \cdot \|\mathcal{D}\|(5\|\mathcal{P}\| + 1)$

where B'_{pmk} is the operating time of the first resource when the kth member resource M is selected. As shown by Figure 1, the operating time of the first resource depends on the surgery time. Thus, we have $B'_{pmk} = G_{p2} + G_{p3} + B_{pMk'} + E_p$. Constraint (87) ensures the length of stay for each patient by each member of resources 2 and 3 are met. Constraint (88) similarly is for patients assigned to resource 1. Constraints (89) and (90) limit the assignment to resources used on the same day. Constraint (91) ensures that at most one patient is assigned to each member at each time unit. Constraints (92), (93), (94), and (95) ensure operations are not preempted. Constraint (96) calculates overtimes. Constraints (97) and (98) define the continuous and binary variables. While perhaps the most popular modeling paradigm for modeling OR scheduling problems, the use of MIP_{TB} is accompanied by many practical and numerical issues. MIP_{TB} requires an extensive time-discretization to remain tractable. In the literature, time discretization of five (Hashemi Doulabi et al., 2016), 15 (Marques et al., 2012) and 30 (Silva et al., 2015) minutes have been reported.

B Size of MIP models

We have captured parametric size dimensionality of our MIPs in terms of the number of variables and constraints in Table 15. We see that MIP_{SB} possesses the lowest size of variables and constraints and MIP_{TB} possesses the highest number of variables and constraints. To further illustrate the size difference in these MIP models, we take their average numbers of variables and constraints when applied to the instances that we have chosen for our experimental analyses (Table 16). Although the number of variables and constraints of a MIP model may not be entirely accurate predictors of a MIP performance (because some MIPs may reach their worst-case complexities), we expect MIP_{SB} to perform better than others. This is due to the more effective trade-off that it can achieve between branching efficiency and bounding effectiveness.

Table 16: Average number of variables and constraints for 8 instances with 25 surgeries, 4 members for each resource.

Model	Variables		Constraints
	Binary	Continuous	
MIP_{SB}	690	1341	15966
MIP_{ISB}	13960	1341	36328
MIP_{PB}	2540	1341	41647
MIP_{TB}	467064	1341	1899430

C Mixed-integer models with step function overtime

In some real-world clinical settings, variable v_{dmk} is considered an integer-valued step function, because overtime is allocated as a Γ-minute block. For example, if an OR requires nurses to stay overtime for one more minute, nurses have to be compensated for, say, $\Gamma = 30$, minutes and if the OR requires 31 minutes of overtime, then compensation has to be 60 minutes. Γ is a discretionary parameter that is determined based on the governing law of each healthcare institution, province, or country. Note, unlike previous models with continuous overtime function with a per-minute cost of overtime, C_{mk} is adjusted accordingly based on the length of Γ in models with step function overtime. The sequence-based MIP_{SB} with step function overtime is denoted as $\text{MIP}_{\text{SB}}^{\text{step}}$ and is as follows:

$$\text{minimize} \quad \sum_{d\in\mathcal{D}}\sum_{m\in\mathcal{M}}\sum_{k\in\mathcal{K}_m}\left(F_{mk}z_{dmk} + \Gamma\cdot C_{mk}u_{dmk}\right) \tag{$\text{MIP}_{\text{SB}}^{\text{step}}$}$$

$$\text{subject to} \quad \text{Constraints (1)} - \text{(19)}$$

$$\Gamma u_{dmk} \geq v_{dmk} \qquad \forall d\in\mathcal{D}, m\in\mathcal{M}, k\in\mathcal{K}_m \tag{99}$$

$$u_{dmk} \in \mathbb{Z}_+ \qquad \forall d\in\mathcal{D}, m\in\mathcal{M}, k\in\mathcal{K}_m \tag{100}$$

where $\mathbb{Z}_+$ denotes the set of non-negative integers. Constraints (99) and the integer variable u_{dmk} in $\text{MIP}_{\text{SB}}^{\text{step}}$ objective function ensure that overtime values are computed in a step function manner. Previous GORPS models, MIP_{ISB} and MIP_{PB}, can be similarly modified to have step function overtimes, which we denote as $\text{MIP}_{\text{ISB}}^{\text{step}}$ and $\text{MIP}_{\text{PB}}^{\text{step}}$, respectively.

D Constraint programming RSSPs

The constraint programming model of RSSPs is as follows.

$$\text{minimize} \quad \left\{C_{mk}\times\max_{p\in\hat{\mathcal{P}}_d^{(i)}}\left(\max\left\{0, \text{EndOf}(\text{Task}_{pmk}) - T\right\}\right)\right\} \tag{RSSP-CP}$$

$$\text{subject to} \quad \text{Constraints (31), (32), and (33)}$$

$$\text{Task}_{pmk} = \text{IntervalVar}(B_{pmk}, \text{Optional}) \qquad \forall p\in\hat{\mathcal{P}}_d^{(i)}, m=\{3\}, k\in\hat{\mathcal{K}}_{mp}^{(i)} \tag{101}$$

$$\text{Task}_{pmk} = \text{IntervalVar}(\text{Optional}) \qquad \forall p\in\hat{\mathcal{P}}_d^{(i)}, m=\{1,2\}, k\in\hat{\mathcal{K}}_{mp}^{(i)} \tag{102}$$

$$\text{Alternative}\left(\text{Task}_{pm}^{*}, \left(\text{Task}_{pmk} : k\in\hat{\mathcal{K}}_{mp}^{(i)}\right)\right) \qquad \forall p\in\hat{\mathcal{P}}_d^{(i)}, m\in\mathcal{M} \tag{103}$$

$$\text{NoOverlap}\left(\text{Task}_{pmk} : p\in\hat{\mathcal{P}}_d^{(i)}\right) \qquad \forall m\in\mathcal{M}, k\in\hat{\mathcal{K}}_m^{(i)} \tag{104}$$

Constraint (101) creates an optional interval variable for each pair of patient and surgeon, eligible to operate the patient. The length of this interval equals to the surgery time by the corresponding surgeon. Constraint (102) creates an optional interval variable for each pair of patient and OR/anesthetist. These two constraints are limited to the allocated members to day d dictated by the AAMP. Constraint (103) re-assigns patients to resources. Constraint (104) avoids overlapping among intervals of each resource member.

E Proofs

E.1 Proof of Proposition 1

Proof. The idea is to divide the total workload by the maximum availability time for a resource member m. The workload of patient p for mth resource is $B_{p3} + Q'_{pm}$. The maximum availability time for a resource member is a function of regular time, overtime, and preparation time (i.e., $T - G'_m + V$). The quantity obtained from this division shows the minimum number of members required from each resource to serve the patients allocated to day d. ■

E.2 Proof of Proposition 2

Proof. We know that a solution is infeasible if the workload of a day (i.e., the patient allocated to the day) is more than capacities (i.e., resource allocated to the day). Thus, this solution is still infeasible unless we lower the workload (i.e., patient removal) or increase/alter the resources. This cut removes any solution in MP for day d that has the same set of patients or more while having the same set of resource or less. For all the solutions with the same or more patients, we have $\sum_{p\in\hat{\mathcal{P}}_d^{(i)}}(1-w_{pd}) = 0$. Hence, it forces MP to alter/add more resources by putting $\sum_{m\in\mathcal{M}}\sum_{k\notin\hat{\mathcal{K}}_{dm}^{(i)}} z_{dmk} \geq 1$. On the other hand, if MP removes any of allocated patients to the day, we have $\sum_{p\in\hat{\mathcal{P}}_d^{(i)}}(1-w_{pd}) \leq 1$. In this case, MP can proceed with the same set of resources, i.e., $\sum_{m\in\mathcal{M}}\sum_{k\notin\hat{\mathcal{K}}_{dm}^{(i)}} z_{dmk} = 0$. ■

E.3 Proof of Proposition 3

Proof. A valid optimality cut removes the current MP sub-optimal solution, but does not remove any feasible integer solutions from the MP feasible region. Assume the incumbent MP solution has allocated set $\hat{\mathcal{P}}_d^{(i)}$ of patients and non-allocated set $\hat{\mathcal{K}}_d^{(i)}$ of resources to day d. Note that all other resources not in $\hat{\mathcal{K}}_d^{(i)}$ are allocated to the day. We define $\mathcal{V}_d^{(i)} = \{\hat{\mathcal{P}}_d^{(i)} \cup \hat{\mathcal{K}}_d^{(i)}\}$. Consider another solution θ with $\mathcal{V}_d^{\theta}$. We either have $\mathcal{V}_d^{(i)} \cap \mathcal{V}_k^{(\theta)} = \mathcal{V}_d^{(i)}$ or $\mathcal{V}_d^{(i)} \cap \mathcal{V}_d^{(\theta)} \neq \mathcal{V}_d^{(i)}$.

Case 1 ($\mathcal{V}_d^{(i)} \cap \mathcal{V}_d^{\theta} = \mathcal{V}_d^{(i)}$). In this case, solution θ contains the same set of patients as the MP incumbent solution $\hat{\mathcal{P}}_d^{(i)}$ or more. It also uses the same set of resources or fewer. In this case, clearly the total overtime cost of solution θ is at least as large as $OFsp_d^{(i)}$. For cut (57), we have

$$\sum_{m\in\mathcal{M}}\sum_{k\in\hat{\mathcal{K}}_{dm}^{(i)}} C_{mk}\cdot v_{dmk} \geq OFsp_d^{(i)}\overbrace{\left(1-\left(\overbrace{\sum_{p\in\hat{\mathcal{P}}_d^{(i)}}(1-w_{pd})+\sum_{m\in\mathcal{M}}\sum_{k\notin\hat{\mathcal{K}}_{dm}^{(i)}} z_{dmk}}^{0}\right)\right)}^{1}$$

Therefore, cut (57) is a valid inequality for solution θ.

Case 2 ($\mathcal{V}_d^{(i)} \cap \mathcal{V}_d^{(\theta)} \neq \mathcal{V}_d^{(i)}$). In this case, solution θ does not include all the patients or/and a different set of resources in $\mathcal{V}_d^{(i)}$; therefore, its total cost may vary. As for cut (57) We have

$$\sum_{m\in\mathcal{M}} \sum_{k\in\hat{\mathcal{K}}_{dm}^{(i)}} C_{mk} \cdot v_{dmk} \geq OFsp_d^{(i)} \overbrace{\left(1 - \left(\overbrace{\sum_{p\in\hat{\mathcal{P}}_d^{(i)}} (1 - w_{pd}) + \sum_{m\in\mathcal{M}} \sum_{k\notin\hat{\mathcal{K}}_{dm}^{(i)}} z_{dmk}}^{\geq 1}\right)\right)}^{\leq 0}$$

Therefore, cut (57) does not impact any of such a solution θ, and this completes the proof. ■

F Data analysis

To achieve higher accuracy and more realistic input for our models, we analyze our data.

F.1 Statistical distributions fitted to our data

There are six parameters (statistical random variables) associated with each surgery: preparation, surgical, removal, PACU, ICU, and ward times. Table 17 shows the average, standard deviation, and skewness of these variables. Preparation, surgical, removal, and PACU durations are expressed in minutes, but ICU and ward duration are expressed in days. We fit five common parametric statistical distributions to these variables and evaluate their goodness-of-fit using the Kolmogorov-Smirnov (KS) test (maximum distance between the cdf of the nominal distribution and the empirical cdf of the data). We skip the 5% tail of sample in order to remove anomalous data that show in the data there are outlier values, e.g., nine months length of stay for some patients in ward beds. We capture KS statistics for each of the five distributions as well as their critical values at $\alpha = 5\%$ to check significance.

Table 17: Distribution of surgery durations. KS statistic is used to determine the best fit; lower values of KS are preferable. Bold indicates the best distribution for each variable.

Variable	Mean	STD	Skew	KS statistic					$\alpha = 5\%$
				Normal	Uniform	Lognormal	Loglogstics	Pearson3	
Prep Time	41	18	0.1	0.096	0.152	0.079	0.650	**0.077**	0.0268
Surgical Time	127	87	1.0	0.159	0.332	**0.049**	0.052	0.077	0.0268
Removal Time	12	4	0.1	0.093	0.176	0.074	0.083	**0.073**	0.0268
PACU Time	85	32	0.1	0.150	0.321	0.760	**0.115**	0.123	0.0285
ICU Time	2.5	1.2	0.4	0.205	0.344	**0.146**	0.147	0.159	0.0458
Ward Time	6.9	6	1.4	0.217	0.591	0.651	0.222	**0.186**	0.0287

Figure 12 shows the designated statistical distributions fit the surgical times, demonstrating that normal and uniform distributions poorly fit the data, whereas the other three distributions better fit the data with the log-normal distribution being the best candidate distribution. The parameters of the fit log-normal distribution to the surgical times are as follows: σ (i.e, shape parameter or standard deviation) is 0.71, θ (i.e., location parameter) is -7.32, and m (i.e, scale parameter or median) is 117.23.

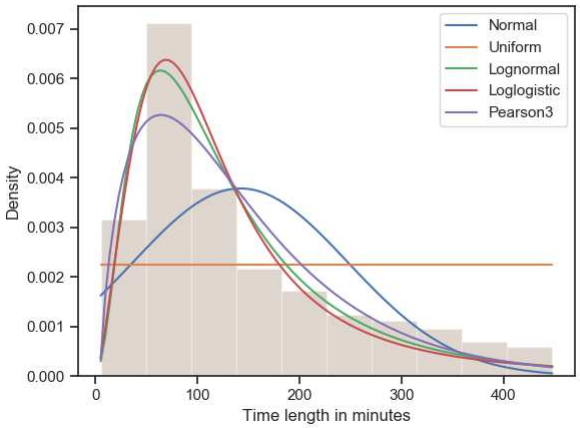


Figure 12: Fitted distributions to the surgical times.

F.2 Correlation among parameters

We investigate whether there is any relationship among the parameters of our model, using a correlation matrix (Table 18). The highest correlation exists among preparation, surgical, and removal times, meaning that as the average value of one of these parameters increases the average value of other parameters also tends to increase.

Table 18: Correlation matrix of parameters

	Preparation	Surgical	Removal	PACU	ICU	Ward
Preparation	1.000	0.472	0.269	0.082	-0.116	0.072
Surgical	0.472	1.000	0.202	0.050	-0.047	0.043
Removal	0.269	0.202	1.000	0.071	0.042	0.076
PACU	0.082	0.050	0.071	1.000	0.041	0.060
ICU	-0.116	-0.047	0.042	0.041	1.000	0.264
Ward	0.072	0.043	0.076	0.060	0.264	1.000

F.3 Possible downstream routes

The most common journey path that patients experience within the operating theatre is

$$\text{OR} \rightarrow \overbrace{\text{PACU} \rightarrow \text{ICU} \rightarrow \text{Ward}}^{\text{downstream units}} \rightarrow \text{Discharge}.$$

Surgeries have variable downstream bed requirements. Some patients are directly discharged from ORs, whereas other patients only stay in some of the downstream units. Out of 2711 surgeries, 2394 patients go to PACU (88.3%). Among these 2075 patients, 319 of them leave the hospital after PACU (11.8%) and do not go to other downstream units. Another 774 patients require both PACU and ICU (28.5%), and all patients going to ICU also will go to a WARD. And lastly, 1301 patients need both PACU and WARD (48%) skipping the ICU. A total of 2075 patients require WARD (76.5%). Figure 13 describes the results.

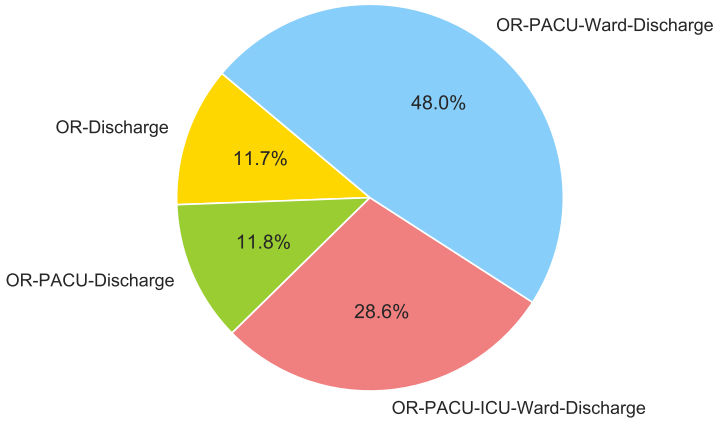


Figure 13: Routes ratios at the downstream units.

F.4 Heterogeneity in the surgical performance of surgeons

There are 166 different surgical services (sub-procedures) offered by the GSD at TGH. The top five most recurring surgeries constitute 42% of all surgeries (Table 19). These 166 sub-procedures are offered by 39 surgeons, 11 of whom (TGH surgeons) perform 90% of surgeries. The GSD of TGH employs surgeons from other specialties within TGH or the same department from Toronto Western Hospital or Princess Margaret Cancer Centre. Each of these invited surgeons operates on less than 3 surgeries in two years, on average. The data reveals that 92 sub-procedures have less than 5 surgeries in two years. Therefore, we only investigate the surgical performance of these 11 surgeons. Table 19 shows that all these 11 surgeons are able to perform surgery type G12580, but only six of them are able to perform surgery type G09870.

Table 19: Statistics for the top-five most frequenting occurring sub-procedures at TGH. Only a subset of

Surgery	Frequency	Percentage	Cumulative	# Surgeons
LAPAROTOMY EXPLORATORY	352	0.13	0.13	11
LAPAROSCOPIC CHOLECYSTECTOMY	234	0.09	0.22	7
LAPAROSCOPIC APPENDECTOMY	228	0.08	0.30	7
WHIPPLE PROCEDURE	200	0.07	0.37	7
RESECTION LIVER RIGHT LOBE	97	0.05	0.42	6

We select the top five most frequent surgeries from Table 19 and perform the Kruskal-Wallis (KW) H test: the non-parametric alternative to the one-way Analysis of Variance test to check the heterogeneity, i.e., significance of surgeons' surgical performance on surgical duration of different surgical sub-procedure. Table 20 shows the average, standard deviation, and coefficient of variation (CV) of surgery times of the five most frequently occurring surgeries in TGH by the four most popular surgeons as well as their p-values. We can conclude that at $\alpha = 0.05$, surgeons' performance significantly impact surgical durations. Additionally, we observe that different surgery types have different levels of variables as captured by their CVs.

Table 20: Statistics for the most frequently occurring surgeries in TGH by the most popular surgeons for these surgeries. Lower p-values are preferable; CV: Coefficient of Variation.

Surgical sub-procedure	General			Surgeons (mean)				p-value
	Mean	STD	CV	1	2	3	4	
LAPAROTOMY EXPLORATORY	128	93	0.73	142	138	181	108	0.05
LAPAROSCOPIC CHOLECYSTECTOMY	80	42	0.53	62	69	82	88	0.00
LAPAROSCOPIC APPENDECTOMY	68	34	0.50	61	67	70	92	0.03
WHIPPLE PROCEDURE	400	101	0.25	361	374	430	420	0.00
RESECTION LIVER RIGHT LOBE	278	108	0.39	283	257	293	275	0.04

F.5 Open scheduling strategy

We allow surgeons to alternate among ORs within a day and operate on their surgeries freely, if it is advantageous from a cost perspective. This practice is called *open scheduling* in the literature. According to our data, 31% of surgeons who have more than one surgery per day visit more than one OR. Implementing an open scheduling policy requires the consideration of sequencing variables and constraints among the starting times of surgeries assigned to ORs and surgeons. Figure 14 shows the movement of a surgeon among ORs.

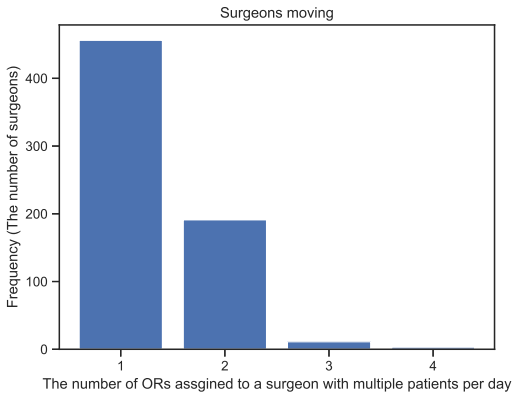


Figure 14: The number of ORs visited by a surgeon with multiple surgeries in a day.

F.6 OR-sharing among specialties

According to our data, from 1011 OR-days, the GSD shares ORs with other departments in 79 OR-days (7.8% of times) in regular OR times (8:00 am to 4:00 pm). OR sharing among specialties is shown to yield a significant downtime for ORs (Marques et al., 2012). We, therefore, do not model this reality for three reasons: (i) its negative impact on OR utilization, (ii) its low probability of occurrence, and (iii) its computational intractability due to considering many specialties and their associated resources.

F.7 Working hours

Figure 15 shows daytime distribution of surgeries. As clearly shown, most surgeries occur between 8:00am to 4:00pm, with a possible extension (overtime, commonly two hours). Having a surgery before or after these thresholds is not common. TGH schedules as many high-priority patients as possible in the OR that

it has solely reserved for emergency patients. For other emergency patients, TGH has a mandate to limit surgeries scheduled over and above regular, daytime hours.

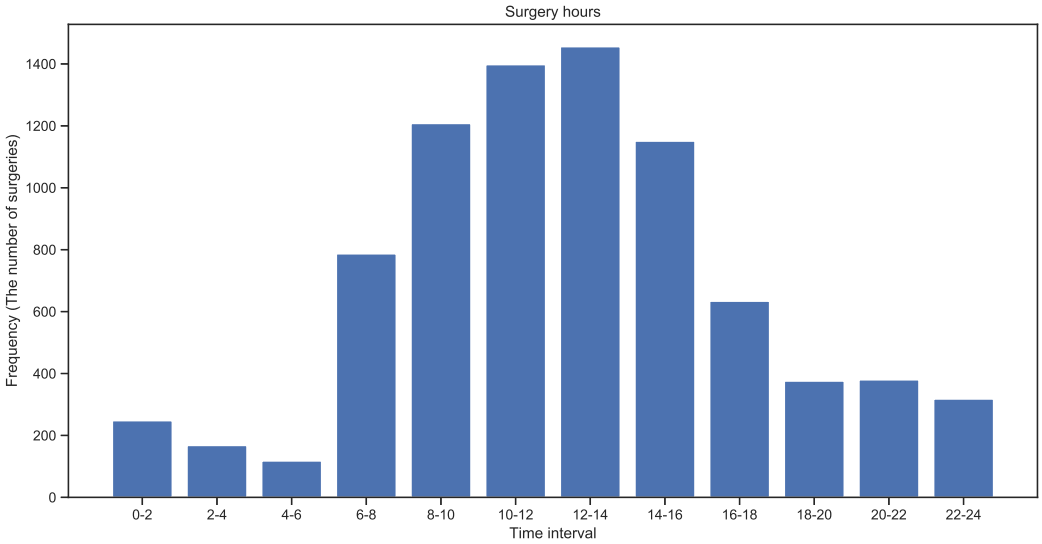


Figure 15: Surgery hours.

F.8 Statistics on number of surgeries

The 2711 surgeries in our data have been *uniformly* distributed over 24 months from July 2011 to June 2013 (Figure 16). The monthly average number of operated patients is approximately 113, and the best-fitted distribution to this data is the Uniform distribution (Figure 16). Given the aggregate OR time allocated to the surgeons in the GSD (obtained from master surgical scheduling phase), this department has always targeted a goal to produce (operate) 110-115 cases per month, and they have almost been successful in achieving their production target. The daily number of surgeries ranges from one to 11.

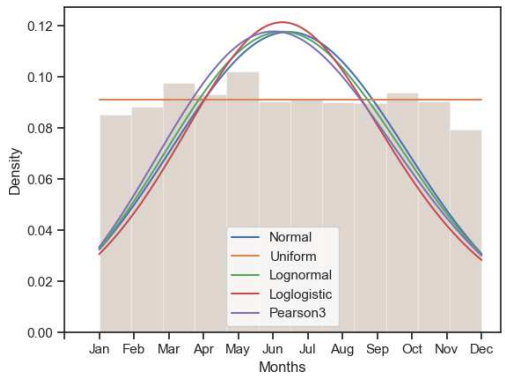


Figure 16: The monthly workload of GSD and distributions of Toronto General Hospital from July 2011 to June 2013.

Figure 17 illustrates the number of days versus the number of opened ORs. It also shows that in most

days, at most four, ORs have been allocated to the GSD. The average number of ORs per day is 2.18. Thus, we consider four ORs is the maximum number of available ORs that they can use in one day.

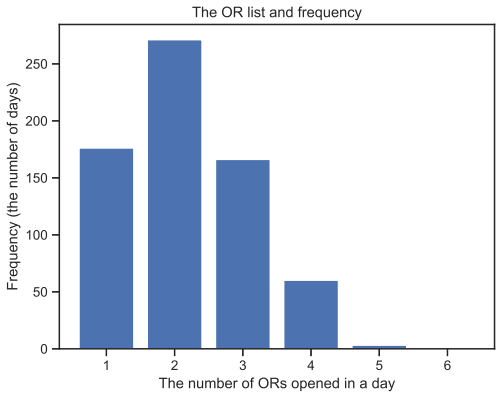


Figure 17: The number of opened OR per day.

References

Augusto, V., X. Xie, V. Perdomo. 2010. Operating theatre scheduling with patient recovery in both operating rooms and recovery beds. *Computers & Industrial Engineering* **58** 231–238.

Barzanji, R., B. Naderi, M. A. Begen. 2019. Decomposition algorithms for the integrated process planning and scheduling problem. *Omega* doi:https://doi.org/10.1016/j.omega.2019.01.003. URL http://www.sciencedirect.com/science/article/pii/S0305048318306698.

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

Beck, J. C. 2010. Checking-up on branch-and-check. David Cohen, ed., *Principles and Practice of Constraint Programming – CP 2010*, *Lecture Notes in Computer Science*, vol. 6308. Springer Berlin Heidelberg, 84–98.

Begen, M. A., M. Queyranne. 2011. Appointment scheduling with discrete random durations. *Mathematics of Operations Research* **36** 240–257.

Belien, J., E. Demeulemeester. 2008. A branch-and-price approach for integrating nurse and surgery scheduling. *European Journal of Operational Research* **189** 652–668.

Bodur, B., S. Dash, O. Günlük, J. Luedtke. 2017. Strengthened Benders cuts for stochastic integer programs with continuous recourse. *INFORMS Journal on Computing* **29** 77–91.

Bodur, M., J. Luedtke. 2017. Mixed-integer rounding enhanced Benders decomposition for multiclass service-system staffing and scheduling with arrival rate uncertainty. *Management Science* **63** 2073–2091.

Bowman, E. H. 1959. The schedule-sequencing problem. *Operations Research* **7** 621–624.

Cardoen, B., E. Demeulemeester, J. Belien. 2009. Sequencing surgical cases in a day-care environment: An exact branch-and-price approach. *Computers & Operations Research* **36** 2660–2669.

Castro, P. M., I. Marques. 2015. Operating room scheduling with generalized disjunctive programming. *Computers & Operations Research* **64** 262–273.

CBCNews. 2017. *Where's Toronto's aging population living? Nearly everywhere in the city, new report finds*. URL https://www.cbc.ca/news/canada/toronto/seniors-report-highlights-demographic-shifts-1.4179023.

Chow, V. S., M. L. Puterman, N. Salehirad, W. Huang, D. Atkins. 2011. Reducing surgical ward congestion through improved surgical scheduling and uncapacitated simulation. *Production and Operations Management* **20** 418–430.

Chu, Y., Q. Xia. 2004. Generating Benders cuts for a general class of integer programming problems. Jean-Charles Régin, Michel Rueher, eds., *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*, *Lecture Notes in Computer Science*, vol. 3011. Springer Berlin Heidelberg, 127–141.

Cire, A. A, W-J van Hoeve. 2013. Multivalued decision diagrams for sequencing problems. *Operations Research* **61** 1411–1428.

COVIDSurg-Collaborative. 2020. Elective surgery cancellations due to the covid-19 pandemic: global predictive modelling to inform surgical recovery plans. *British Journal of Surgery* .

CTVNews. 2020. Provinces begin to address backlog of surgeries in wake of covid-19 URL https://www.ctvnews.ca/health/coronavirus/provinces-begin-to-address-backlog-of-surgeries-in-wake-of-covid-19-1.4932424.

Denton, B. T., A. J. Miller, H. J. Balasubramanian, T. R. Huschka. 2010. Optimal allocation of surgery blocks to operating rooms under uncertainty. *Operations Research* **58** 802–816.

Enayaty-Ahangar, F., C. E. Rainwater, T. C. Sharkey. 2018. A logic-based decomposition approach for multi-period network interdiction models. *Omega* doi:https://doi.org/10.1016/j.omega.2018.08.006. URL http://www.sciencedirect.com/science/article/pii/S0305048318300422.

Fazel-Zarandi, M. M., J. C. Beck. 2012. Using logic-based Benders decomposition to solve the capacity- and distance-constrained plant location problem. *INFORMS Journal on Computing* **24** 387–398.

Fazel-Zarandi, M. M., O. Berman, J. C. Beck. 2013. Solving a stochastic facility location/fleet management problem with logic-based Benders decomposition. *IIE Transactions* **45** 896–911.

Fei, H., C. Chu, N. Meskens, A. Artiba. 2008. Solving surgical cases assignment problem by a branch-and-price approach. *International Journal of Production Economics* **112** 96–108.

Fei, H., N. Meskens, C. Chu. 2010. A planning and scheduling problem for an operating theatre using an open scheduling strategy. *Computers & Industrial Engineering* **58** 221–230.

GlobalNews. 2020. Covid-19 pandemic to affect nearly 400,000 elective surgeries across canada by mid-june: study URL https://globalnews.ca/news/6948692/covid-19-pandemic-elective-surgeries-canada/.

Gul, S., B. T. Denton, J. W. Fowler, T. Huschka. 2011. Bi-criteria scheduling of surgical services for an outpatient procedure center. *Production and Operations Management* **20** 406–417.

Hashemi Doulabi, Hossein, Gilles Pesant, Louis-Martin Rousseau. 2020. Vehicle routing problems with synchronized visits and stochastic travel and service times: Applications in healthcare. *Transportation Science* **54** 1053–1072.

Hashemi Doulabi, S. H., L. M. Rousseau, G. Pesant. 2016. A constraint-programming-based branch-and-price-and-cut approach for operating room planning and scheduling. *INFORMS Journal on Computing* **28** 432–448.

Heching, A., J. N. Hooker, R. Kimura. 2019. A logic-based benders approach to home healthcare delivery. *Transportation Science* doi:10.1287/trsc.2018.0830. URL https://doi.org/10.1287/trsc.2018.0830.

HFMA. 2005. Achieving operating room efficiency through process integration .

Hooker, J. N. 2005. A hybrid method for the planning and scheduling. *Constraints* **10** 385–401. doi:10.1007/s10601-005-2812-2. URL http://dx.doi.org/10.1007/s10601-005-2812-2.

Hooker, J. N. 2007. Planning and scheduling by logic-based Benders decomposition. *Operations Research* **55** 588–602.

Hooker, J. N., G. Ottosson. 2003. Logic-based Benders decomposition. *Mathematical Programming* **96** 33–60.

Jebali, A., A. B. H. Alouane, P. Ladet. 2006. Operating rooms scheduling. *International Journal of Production Economics* **99** 52–62.

Ku, W., J. Christopher Beck. 2016. Mixed integer programming models for job shop scheduling: A computational analysis. *Computers & Operations Research* **73** 165–173.

Ku, W. Y., Thiago. P, J. Christopher Beck. 2014. CIP and MIQP models for the load balancing nurse-to-patient assignment problem. Barry O'Sullivan, ed., *Principles and Practice of Constraint Programming*, *Lecture Notes in Computer Science*, vol. 8656. Springer International Publishing, 424–439.

Laborie, P. 2009. IBM ILOG CP optimizer for detailed scheduling illustrated on three problems. *Integration of AI and OR Techniques in Constraint Programming: 13th International Conference, CPAIOR 2016, Banff, AB, Canada, May 29 - June 1, 2016, Proceedings*. Lecture Notes in Computer Science, Springer Berlin Heidelberg, 148–162.

Manne, A. S. 1960. On the job-shop scheduling problem. *Operations Research* **8** 219–223.

Marques, I., M. E. Captivo, V. P. Margarida. 2012. An integer programming approach to elective surgery scheduling. *OR Spectrum* **34** 407–427. doi:10.1007/s00291-011-0279-7.

Marques, I., M. E. Captivo, V. P. Margarida. 2014. Scheduling elective surgeries in a Portuguese hospital using a genetic heuristic. *Operations Research for Health Care* **3** 59–72.

Naderi, B., A. Azab, K. Borooshan. 2018. A realistic multi-manned five-sided mixed-model assembly line balancing and scheduling problem with moving workers and limited workspace. *International Journal of Production Research* doi:10.1080/00207543.2018.1476786.

Naderi, B., S. M. T. Fatemi Ghomi, M. Aminnayeri, M. Zandieh. 2011. Scheduling open shops with parallel machines to minimize total completion time. *J. Comput. Appl. Math.* **235** 1275–1287.

Naderi, B., V. Roshanaei. 2020. Branch-relax-and-check: A tractable decomposition method for order acceptance and identical parallel machine scheduling. *European Journal of Operational Research* **286** 811 – 827.

Naderi, B., R. Ruiz. 2010. The distributed permutation flowshop scheduling problem. *Computers & Operations Research* **37** 754–768.

Oxley, T.J., Majidi S. Mocco J. 2020. Large-vessel stroke as a presenting feature of COVID-19 in the young. *N Engl J Med* **382** 20. doi:doi:10.1056/NEJMc2009787.

Pan, C.H. 1997. A study of integer programming formulations for scheduling problems. *International Journal of Systems Science* **28** 33–41.

PayScale. 2018. Average anesthesiologist salary. URL https://www.payscale.com/research/CA/Job=Anesthesiologist/Salary.

Perdomo, V., V. Augusto, X. Xie. 2006. Operating theatre scheduling using Lagrangian relaxation. *Service Systems and Service Management, 2006 International Conference on*, vol. 2. 1234–1239.

Pham, D. N., A. Klinkert. 2008. Surgical case scheduling as a generalized job shop scheduling problem. *European Journal of Operational Research* **185** 1011–1025.

Pisinger, D., M. Sigurd. 2007. Using decomposition techniques and constraint programming for solving the two-dimensional bin-packing problem. *INFORMS Journal on Computing* **19** 36–51.

Rahmaniani, Ragheb, Teodor Gabriel Crainic, Michel Gendreau, Walter Rei. 2017. The benders decomposition algorithm: A literature review. *European Journal of Operational Research* **259** 801 – 817. doi:https://doi.org/10.1016/j.ejor.2016.12.005. URL http://www.sciencedirect.com/science/article/pii/S0377221716310244.

Rath, S., K. Rajaram, A. Mahajan. 2017. Integrated anesthesiologist and room scheduling for surgeries: methodology and application. *Operations Research* **65** 1460–1478.

Riise, A., C. Mannino, L. Lamorgese. 2016. Recursive logic-based Benders' decomposition for multi-mode outpatient scheduling. *European Journal of Operational Research* **255** 719–728.

Roshanaei, V., Ahmed Azab, H. ElMaraghy. 2013. Mathematical modelling and a meta-heuristic for flexible job shop scheduling. *International Journal of Production Research* **51** 6247–6274.

Roshanaei, V., K. E.C. Booth, D. M. Aleman, D. R. Urbach, J. C. Beck. 2020a. Branch-and-check methods for multi-level operating room planning and scheduling. *International Journal of Production Economics* **220** 107433.

Roshanaei, V., C. Luong, D. Aleman, D. Urbach. 2017a. Propagating logic-based Benders' decomposition approaches for distributed operating room scheduling. *European Journal of Operational Research* **257** 439–455.

Roshanaei, V., C. Luong, D. M. Aleman, D. Urbach. 2020b. Reformulation, linearization, and decomposition techniques for balanced distributed operating room scheduling. *Omega* **93** 102043.

Roshanaei, V., C. Luong, M. Aleman, D. Urbach. 2017b. Collaborative operating room planning and scheduling. *INFORMS Journal on Computing* **29** 558–580.

Santibanez, P., M. Begen, D. Atkins. 2007. Surgical block scheduling in a system of hospitals: an application to resource and wait list management in a British Columbia health authority. *Health Care Management Science* **10** 269–282.

Silva, T. A. O., M. C. de Souza, R. R. Saldanha, E. K. Burke. 2015. Surgical scheduling with simultaneous employment of specialised human resources. *European Journal of Operational Research* **245** 719 – 730.

Stafford, E. F., F. T. Tseng, J. N. D. Gupta. 2004. Comparative evaluation of MILP flowshop models. *Journal of the Operational Research Society* **56** 88–101.

Thorsteinsson, E. S. 2001. Branch-and-check: A hybrid framework integrating mixed integer programming and constraint logic programming. Toby Walsh, ed., *Principles and Practice of Constraint Programming - CP 2001*, *Lecture Notes in Computer Science*, vol. 2239. Springer Berlin Heidelberg, 16–30.

Tran, T. T, A. Araujo, J. C. Beck. 2016. Decomposition methods for the parallel machine scheduling problem with setups. *INFORMS Journal on Computing* **28** 83–95.

Vijayakumar, B., P. J. Parikh, R. Scott, A. Barnes, J. Gallimore. 2013. A dual bin-packing approach to scheduling surgical cases at a publicly-funded hospital. *European Journal of Operational Research* **224** 583 – 591.

Wagner, H. M. 1959. An integer linear-programming model for machine scheduling. *Naval Research Logistics Quarterly* **6** 131–140.

Wang, T., N. Meskens, D. Duvivier. 2015. Scheduling operating theatres: Mixed integer programming vs. constraint programming. *European Journal of Operational Research* **247** 401 – 413.

Wilson, J.M. 1989. Alternative formulations of a flowshop scheduling problem. *Journal of the Operational Research Society* **40** 395–399.