Logic-based Benders Decomposition and Binary Decision Diagram Based Approaches for Stochastic Distributed Operating Room Scheduling

Cheng Guo*[1], Merve Bodur†[1], Dionne M. Aleman‡[1, 2, 3], and David R. Urbach**[4,5]

[1]Department of Mechanical and Industrial Engineering, University of Toronto, Toronto, Ontario M5S 3G8, Canada

[2]Institute of Health Policy, Management and Evaluation, University of Toronto, Toronto, Ontario M5S 3E3, Canada

[3]Techna Institute at University Health Network, Toronto, Ontario M5G 1P5, Canada

[4]Department of Surgery, Women's College Hospital, Toronto, Ontario M5S 1B2, Canada

[5]Department of Surgery, University of Toronto, Toronto, Ontario M5S 3E3, Canada

Abstract

The distributed operating room (OR) scheduling problem aims to find an assignment of surgeries to ORs across collaborating hospitals that share their waiting lists and ORs. We propose a stochastic extension of this problem where surgery durations are considered to be uncertain. In order to obtain solutions for the challenging stochastic model, we use sample average approximation, and develop two enhanced decomposition frameworks that use logic-based Benders (LBBD) optimality cuts and binary decision diagram based Benders cuts. Specifically, to the best of our knowledge, deriving LBBD optimality cuts in a stochastic programming context is new to the literature. Our computational experiments on a hospital dataset illustrate that the stochastic formulation generates robust schedules, and that our algorithms improve the computational efficiency.

1 Introduction

According to a recent report by the Canadian Institute for Health Information, about 30% of patients who need to receive hip replacement, knee replacement, and cataract surgeries wait longer than the recommended waiting time (CIHI, 2020). As a result of such lengthy waiting times for medically necessary treatments, patients suffer from lost wages and reduced productivity, due to

*cguo@mie.utoronto.ca

†bodur@mie.utoronto.ca

‡aleman@mie.utoronto.ca

**david.urbach@wchospital.ca

the effects of an untreated medical condition on mind and body (Barua and Jacques, 2019). The research on operating room (OR) scheduling studies how to allocate OR resources more strategically to improve its utilization. In this work, we study the *stochastic distributed OR scheduling (SDORS) problem*, where several hospitals share their surgery waiting lists and ORs, and collectively schedule patients. We acknowledge the stochasticity in surgery procedures and derive schedules that remain robust under such an uncertainty.

Studies from both hospital practice and mathematical modeling demonstrate that by utilizing distributed OR scheduling (DORS), ORs see an increased utilization rate (Magnussen et al., 2007; Wang et al., 2016). This is because when hospitals collaborate in waiting list management, they achieve a more balanced schedule across hospitals.

On the other hand, including multiple hospitals in the OR scheduling process adds to its uncertainty, such as the uncertainty in the surgery durations. This may result in an increased number of surgery cancellations, which can be costly for both the cancelled patients and the hospital. In this paper, we consider day of the surgery cancellations due to OR overtime. Lack of OR time can be a significant contributor to same-day cancellations; for instance Dimitriadis et al. (2013) found that lack of OR time caused 17.31% of all same-day cancellations.

In order to derive more robust schedules in the face of such a stochasticity, we use stochastic optimization. More specifically, we model SDORS as a two-stage stochastic integer program (2SIP), which is widely used in planning and scheduling problems. In our problem we have continuous random variables in the 2SIP, and such models are usually reformulated via sample average approximation (SAA). However, mostly due to integer decision variables in both stages, solving the SAA reformulation directly with commercial solvers can take a prohibitively long time, especially when the instance size grows. Furthermore, the presence of integer variables, particularly in the second stage, limits the choice of decomposition algorithms that can be employed, and highly impact the algorithmic efficiency.

In this work, we develop two decomposition algorithms that are applicable when there are integer variables in the decomposed subproblems. Those decomposition algorithms use logic-based Benders (LBBD) cuts and Binary Decision Diagram (BDD) based Benders cuts. We also incorporate classical Benders cuts from the linear programming (LP) relaxations of subproblems, which helps to improve the convergence. Moreover, we propose several algorithmic enhancements, including adapted first fit decreasing (FFD) heuristics to find initial solutions, relaxations of the subproblems that are used to tighten the master problem, and an early stopping scheme to eliminate suboptimal master solutions faster, which significantly improve computational efficiency.

The contributions of this work lies both in the modeling and the solution methodology. In the modeling side, we propose the SDORS model, and evaluate its value against the DORS model from the literature (Roshanaei et al., 2017) . In the methodology aspect, we apply LBBD on a 2SIP and derive LBBD optimality cuts. To our best knowledge, the application of LBBD optimality cuts in stochastic programming is new to the literature. We also show the efficacy of the BDD-based decomposition algorithm, proposed by Lozano and Smith (2018), for a new problem class, while in the literature it was tested only on the traveling salesman problem with time windows. Note

that SDORS is a special type of planning and scheduling problem. Therefore, the decomposition algorithms we derive for SDORS are also applicable for other planning and scheduling models that share similar structures.

The rest of this paper is organized as follows: Section 2 provides a review of the literature on OR scheduling problems and 2SIP algorithms, also detailing some important concepts used in our work. Section 3 presents a detailed description of the SDORS problem and its formulation. Section 4 explains the SAA framework. Section 5 and Section 6 introduce a two-stage and a three-stage decomposition algorithm, respectively, to solve the SAA problem. Finally, Section 7 presents the experimental results.

2 Literature Review

The OR scheduling problem, which involves different types of planning decisions, studies the allocation of resources related to surgeries. In this work, we include both hospital and OR opening decisions as well as surgery scheduling decisions, under the uncertainty of surgery durations. Extensive literature reviews on OR scheduling can be found in Cardoen et al. (2010) and Guerriero and Guido (2011). We focus our review on objective and decision variable setups, and on stochastic OR scheduling models.

The OR scheduling is well-studied in the *deterministic setting*, where we assume all parameters are known. A variety of objectives are used in the scheduling models. Marques et al. (2012) propose a mixed-integer programming (MIP) model that both improves the OR occupancy rate and shortens the surgery waiting time. Fei et al. (2008) reduce the underutilization and overtime of ORs. Roshanaei et al. (2017) minimize the hospital and OR opening costs, and try to schedule patients based on their priority scores. Other goals of OR scheduling include improving the throughput (Harper, 2002), avoiding peaks in resource occupancy (Beliën and Demeulemeester, 2008), and incorporating preferences of different parties (Blake and Carter, 2002). OR scheduling models in the literature also include different types of decisions. Blake and Donald (2002) assign ORs to different specialties in a hospital. Beliën and Demeulemeester (2008) study the assignment of surgeons to ORs. Jebali et al. (2006) consider both the assignment of patients to ORs and the sequencing of patiens in an OR. Similar to Roshanaei et al. (2017), our model makes opening decisions and patient-to-OR assignment decisions, and our objective is to minimize the opening costs while accounting for patients' priority scores. In addition, we include cancellation decisions for overtime surgeries, and try to reduce surgery cancellations.

In real life, the scheduling of ORs can be affected by *uncertainty* in surgery procedures. By considering the uncertainty in OR scheduling, we can generate a more robust surgery schedule and achieve cost savings (Min and Yih, 2010). Uncertainties in different stages of the process are considered in the literature. Denton et al. (2010) minimize the cost in an OR scheduling problem assuming stochastic overtime. Deng et al. (2019) deal with uncertainty in both waiting times and surgery durations. Bowers and Mould (2004) consider the uncertainty in patient arrivals. Also, stochastic OR scheduling problems in the literature use a variety of modeling techniques,

including 2SIP (Denton et al., 2010; Min and Yih, 2010), robust optimization (Denton et al., 2010), distributionally robust chance constraints (Deng et al., 2019), and multi-stage stochastic programming (Gul et al., 2015). In our work, we formulate a 2SIP model which incorporates the stochasticity from surgery durations.

The importance of *distributed* planning and scheduling gained attention in many industries including manufacturing and healthcare. In a distributed setting, several agents share their resources and the list of tasks. Distributed manufacturing has the advantage of improving product quality, reducing costs and management risks, which helps the manufacturer to become more competitive under globalization (Naderi and Ruiz, 2014). Timpe and Kallrath (2000) provide MIP formulations to solve lot-sizing problems in a distributed setting. Behnamian and Ghomi (2013) study a multi-factory production problem to minimize the maximum makespan. Inspired by the practice of hospitals in Toronto, Roshanaei et al. (2017) propose a MIP model for the DORS problem where hospitals share their waiting lists and ORs. They find that in the distributed setting, hospitals are able to significantly reduce costs thanks to increased patient admission rate and OR throughput. For a similar setting, Roshanaei et al. (2017) propose an LBBD algorithm for DORS to solve practical instances efficiently. Our work is closely related to Roshanaei et al. (2017) and Roshanaei et al. (2017), as we also study the OR scheduling problem in a distributed setting. However, those previous works do not consider uncertainty in surgery durations, thus their proposed schedules may result in cancellations due to surgeries exceeding OR operating time limits. Moreover, our proposed SDORS model is significantly more challenging to solve, which calls for a thorough investigation of the problem structure to design efficient algorithms.

SAA is the most common approach for 2SIPs, where the SAA problem, also known as the *deterministic equivalent* (DE), is usually solved exactly by means of *decomposition* dividing the problem into a master problem and a subproblem. Benders decomposition is one of the most commonly-used decomposition algorithms (Benders, 1962). However, it is not applicable when the subproblem contains integer variables. Laporte and Louveaux (1993) propose the integer L-Shaped method for 2SIPs, which allows the subproblem to contain integer variables. Angulo et al. (2016) present an improved integer L-Shaped method. Carøe and Tind (1998) propose a generalized Benders decomposition method for 2SIP, which might generate nonlinear cuts and is computationally expensive in general cases. Sherali and Fraticelli (2002) use the Reformulation-Linearization Technique to generate cuts from integer recourse problems. Gade et al. (2014) use parametric Gomory cuts in their decomposition algorithm, which shows effectiveness for a general class of 2SIPs. Some other cutting planes for 2SIPs with discrete recourse include the lift-and-project cuts (Balas et al., 1993; Ntaimo and Tanner, 2008), Fenchel decomposition (FD) cuts (Ntaimo, 2013), and scenario FD cuts (Beier et al., 2015). For a more detailed review on exact decomposition methods for 2SIPs, we refer the readers to Bodur (2015). In this work, we instead choose to use LBBD and BDD-based methods, to benefit from our specific problem structure.

The *LBBD method* proposed by Hooker and Ottosson (2003) provides a general way to decompose the problem such that the master problem and subproblem can be of any structure. Similar to the classical Benders method, the LBBD method generates LBBD feasibility cuts when the

subproblem is infeasible, while LBBD optimality cuts are generated when the master problem underestimates the subproblem objective. However, both types of LBBD cuts are problem-specific, and their strength heavily depends on the incorporation of the underlying structure information. When strong cuts are derived, LBBD can be much more efficient than generic 2SIP methods. In some recent works, LBBD is used to solve large-scale 2SIPs. Lombardi et al. (2010) solve a stochastic allocation and scheduling problem for a multi-processor system-on-chips problem with LBBD. Fazel-Zarandi et al. (2013) propose an LBBD method for a facility location/fleet management problem. Those papers formulate *only LBBD feasibility cuts* in the decomposition. In our work, we propose *LBBD optimality cuts* for our 2SIP SDORS model in two different decomposition frameworks.

Our proposed decomposition algorithms for SDORS also use *BDD-based cuts*. BDD is a graph structure that transforms a binary integer program (BIP) to an LP using a recursive formulation of the original problem. Lozano and Smith (2018) propose a decomposition algorithm based on BDD to solve a special class of 2SIPs, where each of the constraints linking the first and second stages consists of some binary second stage variables and a single first stage variable, and is deactivated when the corresponding first stage variable is zero. They transform the recourse problem to a capacitated shortest path problem, which has an LP formulation thus leads to a classical Benders decomposition algorithm. They conduct numerical experiments on the traveling salesman problem with time windows and show that their algorithm achieves substantial speedups compared to a commercial IP solver. In our work, we adapt an enhanced version of their decomposition algorithm to solve a 2SIP in the distributed OR scheduling setting.

3 Problem Definition and Formulation

In the SDORS problem, we aim to assign surgeries to the ORs in collaborative hospitals in the current planning horizon. The surgery durations are assumed to be stochastic. For each OR, there is a daily operating time limit. If the total surgery time of the day is expected to exceed the limit, some of the surgeries will be cancelled to satisfy this constraint. Our goal is to minimize total costs while ensuring more emergent patients get scheduled first.

We formulate the SDORS problem as a 2SIP. We assume that the probability distributions of surgery durations are known. In a 2SIP the decision process is divided into two stages. The first stage problem is solved before the revelation of surgery durations, where we make the decisions of (1) which surgical suite to open during the current planning horizon, (2) which ORs to open in an opened hospital, (3) which patients to assign to the opened ORs, and (4) which patients to postpone to a future planning horizon. Notice that we consider only one list of patients in our model, for instance those patients may be from a single specialty, and we consider a surgical suite open if it is going to be (partially) used for our targeted patient list.

In this paper, we consider day of the surgery cancellations. A 2013 study (Dimitriadis et al., 2013) shows that the same-day surgery cancellation rate in the study is 5.19%. Among all same-day cancellations, 17.31% are due to lack of theatre time. We assume each OR has an operating time

limit for the total duration of the surgeries during a day, any surgery that is expected to finish after that time limit will be cancelled before it starts. The cancelled surgeries need to be scheduled in a future planning horizon instead. Therefore, each cancellation is associated with some costs. The second stage of the 2SIP is a recourse problem that takes the surgery-OR assignment decisions from the first stage, and decides on which surgeries to cancel, while minimizing the total cancellation cost. Notice that 2SIP models implicitly assume that in the second stage, all surgery durations are known with certainty at the same time. In addition, in real life cancellation decisions rely on the sequence of surgeries, which we do not consider in our model. We note that the implementable decisions from our model are only the first-stage ones, i.e., opening and assignment decisions. By considering surgery durations and cancellations in different scenarios, we are able to create more flexible schedules than the deterministic models, and reduce the occurrences of cancellations. Incorporation of surgery sequencing will make our model more realistic and possibly yield better schedules. However, the joint optimization of our current decisions and the sequencing ones would become computationally very difficult. Therefore, we leave it for future research and propose our current model as a first step in showing the importance of incorporating surgery duration uncertainty to the *distributed* OR scheduling problems. We also observe that many stochastic scheduling models (in a variety of application domains) make the simplifying assumption to ignore sequencing decisions, such as Fei et al. (2009), Gul et al. (2015) and Bodur and Luedtke (2016). In particular, similar to our paper, Gul et al. (2015) assume no sequencing and all surgery durations are revealed at the same time. Their results show decreased costs compared with deterministic models. Therefore, we believe our model is still helpful in improving the quality of distributed OR scheduling under uncertainty, despite having some simplifying assumptions.

Except for the stochasticity of surgery durations and cancellation decisions, all other settings in the SDORS model are the same as in the DORS model of Roshanaei et al. (2017). More specifically, for hospitals in the set $\mathcal{H}$ we need to decide which patients to schedule in the current planning horizon $\mathcal{D}$, which is a set of days. Each hospital h has a set of ORs, $\mathcal{R}_h$. For simplicity we assume that all ORs in the set $\mathcal{R}_h$ are homogenous, and all hospitals have the same number of ORs, none of which is a limiting assumption. For example, in real life ORs could be nonidentical, which means some surgeries can only be practiced in a subset of all available ORs. This type of requirement can be modelled by extra constraints such as in Roshanaei et al. (2017). Each patient in the set $\mathcal{P}$ has a health status score ω_p. Given a health score threshold Γ, all patients whose health score $\omega_p \geq \Gamma$ are called *mandatory patients* and denoted by the set $\mathcal{P}'$, and the rest of the patients are defined as *non-mandatory patients*. Mandatory patients have to be scheduled in the current planning horizon, while non-mandatory patients can be postponed to a future planning horizon. The objective is to minimize the total cost, which we will explain in more details later in this section.

The notations we use in the model are listed in Table 1.

Table 1: Notation

Sets:

$\mathcal{P}$	Set of patients, $p \in \mathcal{P}$
$\mathcal{P}'$	Set of mandatory patients
$\mathcal{H}$	Set of hospitals, $h \in \mathcal{H}$
$\mathcal{D}$	Set of days in the current planning horizon, $d \in \mathcal{D}$
$\mathcal{R}_h$	Set of ORs in the surgical suite of hospital h, $r \in \mathcal{R}_h$
$\mathcal{S}$	Set of possible scenarios of uncertain surgery durations, $s \in \mathcal{S}$
Parameters:	
G_{hd}	Cost of opening the surgical suite of hospital h in day d
F_{hd}	Cost of opening an OR in hospital h on day d
B_{hd}	Operating time limit of each OR on day d in hospital h
T_p	Total booked time of patient p
c_{dp}^{sched}	Benefit for scheduled patient p whose surgery is scheduled on day d
c_p^{unsched}	Penalty for unscheduled patient p
c_p^{cancel}	Penalty for cancelled patient p
Decision variables:	
u_{hd}	1 if the surgical suite in hospital h is opened on day d, 0 otherwise
y_{hdr}	1 if OR r of hospital h is opened on day d, 0 otherwise
x_{hdpr}	1 if patient p is assigned to OR r of hospital h on day d, 0 otherwise
w_p	1 if patient p is postponed to a future planning horizon, 0 otherwise
z_{hdpr}	1 if patient p's surgery in OR r of hospital h on day d is operated, 0 if it is cancelled

The SDORS problem is formulated as follows:

$$\text{(SDORS):}\ \min \sum_{h\in\mathcal{H}}\sum_{d\in\mathcal{D}} G_{hd}u_{hd} + \sum_{h\in\mathcal{H}}\sum_{d\in\mathcal{D}}\sum_{r\in\mathcal{R}_h} F_{hd}y_{hdr} + \sum_{h\in\mathcal{H}}\sum_{d\in\mathcal{D}}\sum_{p\in\mathcal{P}}\sum_{r\in\mathcal{R}_h} c_{dp}^{\text{sched}}x_{hdpr} + \sum_{p\in\mathcal{P}\setminus\mathcal{P}'} c_p^{\text{unsched}}w_p + \mathbb{E}_{\mathbf{T}}\mathcal{Q}(\mathbf{x},\mathbf{y},\mathbf{T}) \tag{1a}$$

$$\text{s.t.} \sum_{h\in\mathcal{H}}\sum_{d\in\mathcal{D}}\sum_{r\in\mathcal{R}_h} x_{hdpr} = 1 \qquad \forall p \in \mathcal{P}' \tag{1b}$$

$$\sum_{h\in\mathcal{H}}\sum_{d\in\mathcal{D}}\sum_{r\in\mathcal{R}_h} x_{hdpr} + w_p = 1 \qquad \forall p \in \mathcal{P}\setminus\mathcal{P}' \tag{1c}$$

$$y_{hdr} \le y_{hd,r-1} \qquad \forall h\in\mathcal{H}, d\in\mathcal{D}, r\in\mathcal{R}_h\setminus\{1\} \tag{1d}$$

$$\sum_{p\in\mathcal{P}} c_p^{\text{cancel}}x_{hdpr} \le \sum_{p\in\mathcal{P}} c_p^{\text{cancel}}x_{hdp,r-1} \qquad \forall h\in\mathcal{H}, d\in\mathcal{D}, r\in\mathcal{R}_h\setminus\{1\} \tag{1e}$$

$$y_{hdr} \le u_{hd} \qquad \forall h\in\mathcal{H}, d\in\mathcal{D}, r\in\mathcal{R}_h \tag{1f}$$

$$x_{hdpr} \le y_{hdr} \qquad \forall h\in\mathcal{H}, d\in\mathcal{D}, p\in\mathcal{P}, r\in\mathcal{R}_h \tag{1g}$$

$$u_{hd}, y_{hdr}, x_{hdpr}, w_p \in \{0,1\} \qquad \forall h\in\mathcal{H}, d\in\mathcal{D}, p\in\mathcal{P}, r\in\mathcal{R}_h \tag{1h}$$

where

$$\mathcal{Q}(\mathbf{x},\mathbf{y},\mathbf{T}) = \min \sum_{h\in\mathcal{H}}\sum_{d\in\mathcal{D}}\sum_{p\in\mathcal{P}}\sum_{r\in\mathcal{R}_h} c_p^{\text{cancel}}(x_{hdpr} - z_{hdpr}) \tag{2a}$$

$$\text{s.t.} \sum_{p\in\mathcal{P}} T_p z_{hdpr} \le B_{hd}y_{hdr} \qquad \forall h\in\mathcal{H}, d\in\mathcal{D}, r\in\mathcal{R}_h \tag{2b}$$

$$z_{hdpr} \leq x_{hdpr} \qquad \forall h \in \mathcal{H}, d \in \mathcal{D}, p \in \mathcal{P}, r \in \mathcal{R}_h \qquad (2c)$$

$$z_{hdpr} \in \{0,1\} \qquad \forall h \in \mathcal{H}, d \in \mathcal{D}, p \in \mathcal{P}, r \in \mathcal{R}_h \qquad (2d)$$

representing the optimization problem that minimizes the second stage cancellation cost, parameterized by the first stage decisions $\mathbf{x} = \{x_{hdpr} : \forall h \in \mathcal{H}, d \in \mathcal{D}, p \in \mathcal{P}, r \in \mathcal{R}_h\}$, $\mathbf{y} = \{y_{hdr} : \forall h \in \mathcal{H}, d \in \mathcal{D}, r \in \mathcal{R}_h\}$, and $\mathbf{T} = \{T_1, ..., T_{|\mathcal{P}|}\}$. Note that $\mathbf{T}$ is the set of given parameters for surgery durations.

The first stage objective function (1a) minimizes the total operational and expected cancellation costs, whose terms are respectively (i) the total cost of opening surgical suites in hospitals, (ii) the total cost of opening ORs, (iii) the benefit of scheduling patients (note that this term is negative), (iv) the penalty for postponing patients, and (v) the expected total cancellation cost. Constraints (1b) ensure that mandatory patients are all scheduled in the current planning horizon. Constraints (1c) either schedule non-mandatory patients in the current planning horizon or postpone their surgeries to a future time. Constraints (1d) are symmetry breaking constraints for ORs in the same hospital, since they are homogeneous. Constraints (1e) are symmetry breaking constraints ensuring that only one permutation of patient assignment is allowed in each hospital-day pair (h, d). More specifically, these constraints require that for ORs in the same hospital, the total cancellation cost of patients in an OR should be higher if this OR has a lower index. These constraints break more symmetry if cancellation costs are different for different patients, which is generally the case in our data. Constraints (1f) link u and y variables, to ensure that if the surgical suite in a hospital is closed, no OR in it is opened. Similarly, constraints (1g) make sure that a patient will only be assigned to an opened OR. Lastly, constraints (1h) enforce binary restrictions on the decision variables.

For the second stage, i.e., the recourse problem (2), constraints (2b) ensure that the surgeries which are eventually conducted in an OR will not exceed its operating time limit. Constraints (2c) link variables z and x, which makes sure a patient's surgery can only be performed if it is scheduled in the first stage. Note that the second stage problem is always feasible, regardless of the first stage solution, because in the worst case we can cancel all the scheduled patients. Therefore, (SDORS) has *complete recourse*.

4 The SAA Problem

Solving (SDORS) exactly involves calculating the multidimensional expectation over a set of continuous random vector $\mathbf{T}$. To overcome this difficulty, we solve the problem approximately by SAA. We generate a set of *scenarios* for the surgery duration vector $\mathbf{T}$. The set of scenarios are denoted by $\mathcal{S}$ and the surgery duration of patient p under scenario $s \in \mathcal{S}$ is denoted by T_p^s. Under each scenario we need to decide whether to operate or cancel a surgery, thus we now have a decision variable z_{hdpr}^s for each scenario that represents such decisions. We assume that all scenarios are equally likely, so each has a realization probability of $1/|\mathcal{S}|$.

Replacing the expected cancellation cost with the average cancellation cost over the scenarios,

we obtain the DE of the 2SIP:

$$\text{(DE)}: \min \quad \sum_{h\in\mathcal{H}}\sum_{d\in\mathcal{D}} G_{hd}u_{hd} + \sum_{h\in\mathcal{H}}\sum_{d\in\mathcal{D}}\sum_{r\in\mathcal{R}_h} F_{hd}y_{hdr} + \sum_{h\in\mathcal{H}}\sum_{d\in\mathcal{D}}\sum_{p\in\mathcal{P}}\sum_{r\in\mathcal{R}_h} c_{dp}^{\text{sched}}x_{hdpr} +$$
$$\sum_{p\in\mathcal{P}\setminus\mathcal{P}'} c_p^{\text{unsched}}w_p + \frac{1}{|\mathcal{S}|}\sum_{s\in\mathcal{S}}\sum_{h\in\mathcal{H}}\sum_{d\in\mathcal{D}}\sum_{p\in\mathcal{P}}\sum_{r\in\mathcal{R}_h} c_p^{\text{cancel}}(x_{hdpr} - z_{hdpr}^s) \tag{3a}$$

$$\text{s.t.} \quad \text{(1b)} - \text{(1h)} \tag{3b}$$

$$\sum_{p\in\mathcal{P}} T_p^s z_{hdpr}^s \le B_{hd}y_{hdr} \qquad \forall h\in\mathcal{H}, d\in\mathcal{D}, r\in\mathcal{R}_h, s\in\mathcal{S} \tag{3c}$$

$$z_{hdpr}^s \le x_{hdpr} \qquad \forall h\in\mathcal{H}, d\in\mathcal{D}, p\in\mathcal{P}, r\in\mathcal{R}_h, s\in\mathcal{S} \tag{3d}$$

$$z_{hdpr}^s \in \{0,1\} \qquad \forall h\in\mathcal{H}, d\in\mathcal{D}, p\in\mathcal{P}, r\in\mathcal{R}_h, s\in\mathcal{S} \tag{3e}$$

Next, we present two decomposition schemes to solve this model exactly.

5 Two-stage Decomposition

In this section we first introduce the decomposition scheme that divides the (DE) into a master problem and a set of subproblems, where each subproblem is a BIP. We generate three types of cutting planes from the subproblems: LBBD optimality cuts, classical Benders cuts from their BDD reformulations (BDD-based Benders cuts), and classical Benders cuts from their LP relaxations. Notice that the first two cut families are exact, meaning solely adding violated cuts from one such family will lead the algorithm to converge to an optimal (DE) solution. We also introduce several algorithmic enhancements for the decomposition, some of which are novel to the literature.

5.1 Decomposition Framework

We decompose our problem into a master problem and a set of subproblems. In the master problem the first-stage decisions, namely hospital opening, OR opening, patient assignment, and postponing decisions, are made. In the subproblems we make decisions about surgery cancellation. More specifically, the master problem contains the variables u_{hd}, y_{hdr}, x_{hdpr}, and w_p from the original problem, and a new variable Q_{hdr}^s representing the cancellation cost of the hospital-day-OR combination (h,d,r) under scenario s:

$$\min \quad \sum_{h\in\mathcal{H}}\sum_{d\in\mathcal{D}} G_{hd}u_{hd} + \sum_{h\in\mathcal{H}}\sum_{d\in\mathcal{D}}\sum_{r\in\mathcal{R}} F_{hdr}y_{hdr} + \sum_{h\in\mathcal{H}}\sum_{d\in\mathcal{D}}\sum_{p\in\mathcal{P}}\sum_{r\in\mathcal{R}} c_{dp}^{\text{sched}}x_{hdpr}$$
$$+ \sum_{p\in\mathcal{P}\setminus\mathcal{P}'} c_p^{\text{unsched}}w_p + \frac{1}{|\mathcal{S}|}\sum_{s\in\mathcal{S}}\sum_{h\in\mathcal{H}}\sum_{d\in\mathcal{D}}\sum_{r\in\mathcal{R}} Q_{hdr}^s \tag{4a}$$

$$\text{s.t.} \quad \text{(1b)} - \text{(1h)} \tag{4b}$$

$$\text{[LBBD cuts or BDD-based Benders cuts]} \tag{4c}$$

$$Q_{hdr}^s \ge 0, \qquad \forall s\in\mathcal{S}, h\in\mathcal{H}, d\in\mathcal{D}, r\in\mathcal{R}_h \tag{4d}$$

Constraints (4c) are BDD-based Benders cuts or LBBD cuts that are generated progressively from the subproblems, in order to correctly approximate the second-stage value function, which is the link between the Q variables and the first-stage variables. Without cutting planes (4c), the master problem (4) is a relaxation of (DE). The value of cancellation cost Q^s_{hdr} is underestimated because the master problem lacks enough information regarding the recourse decisions. On the other hand, subproblems contain the information about the recourse decisions. We provide such information by generating cutting planes from subproblems and adding them to the master problem.

By solving the master problem, we get the optimal solutions for u_{hd}, y_{hdr}, x_{hdpr}, w_p and Q^s_{hdr}, denoting their optimal solutions respectively by $\hat{u}_{hd}$, $\hat{y}_{hdr}$, $\hat{x}_{hdpr}$, $\hat{w}_p$, and $\hat{Q}^s_{hdr}$. Then we pass optimal solutions of x_{hdpr} and y_{hdr} to the subproblems. Note that we formulate a subproblem for each (h, d, r, s) tuple, as the recourse decisions for each OR and scenario is independent once the values of x_{hdpr} and y_{hdr} are fixed. Each subproblem minimizes the cancellation cost Q^s_{hdr} by selecting the least costly patients to cancel, if the current assignment exceeds the operating time limit of an OR:

$$\mathcal{Q}^s_{hdr}(\hat{x}_{hd\cdot r}, \hat{y}_{hdr}, T^s_{\cdot}) = \min \quad \sum_{p\in\mathcal{P}} c^{\text{cancel}}_p(\hat{x}_{hdpr} - z^s_{hdpr}) \tag{5a}$$

$$\text{s.t.} \quad \sum_{p\in\mathcal{P}} T^s_p z^s_{hdpr} \leq B_{hd} \tag{5b}$$

$$z^s_{hdpr} \leq \hat{x}_{hdpr} \qquad \forall p \in \mathcal{P} \tag{5c}$$

$$z^s_{hdpr} \in \{0, 1\} \qquad \forall p \in \mathcal{P} \tag{5d}$$

Note that we use $\cdot$ to represent all members of the index in that position, thus $\hat{x}_{hd\cdot r} = \{\hat{x}_{hd1r}, ..., \hat{x}_{hd|\mathcal{P}|r}\}$ and $T^s_{\cdot} = \{T^s_1, ..., T^s_{|\mathcal{P}|}\}$. Constraints (5b) are the time limit constraints and constraints (5c) guarantee that no surgery will be performed if it is not scheduled in the first place. Constraints (5d) enforce variables z^s_{hdpr} to be binary. To simplify the notation, from now on we use $\bar{Q}^s_{hdr}$ to denote $\mathcal{Q}^s_{hdr}(\hat{x}_{hd\cdot r}, \hat{y}_{hdr}, T^s_{\cdot})$, which is the correct optimal value of the subproblem objective.

5.2 The LBBD cuts

The first category of cutting planes that we derive are the LBBD cuts. Like the case with classical Benders cuts, there are two types of LBBD cuts: LBBD feasibility cuts and LBBD optimality cuts. LBBD feasibility cuts are generated when the master problem solution makes a subproblem infeasible. On the other hand, LBBD optimality cuts are generated when the master problem solution underestimates the (minimization) objective value of a feasible subproblem, due to lacking the information about the subproblem. In our decomposition framework, the subproblems are never infeasible, since it can always cancel all the assigned patients and get a feasible solution. As such, we do not need LBBD feasibility cuts.

Since the master problem is a relaxation of (DE), the cancellation cost $\hat{Q}^s_{hdr}$ is an underestimate of the optimal cancellation cost for the assignment $(\hat{u}_{hd}, \hat{y}_{hdr}, \hat{x}_{hdpr}, \hat{w}_p)$ from the master problem, while the subproblem objective $\bar{Q}^s_{hdr}$ provides the actual cancellation cost of such an assignment.

When $\hat{Q}^s_{hdr} < \bar{Q}^s_{hdr}$, we add the following LBBD optimality cut to the master problem:

$$Q^s_{hdr} \geq \bar{Q}^s_{hdr} - \sum_{p \in \hat{\mathcal{P}}_{hdr}} c^{\text{cancel}}_p (1 - x_{hdpr}) \tag{6}$$

where $\hat{\mathcal{P}}_{hdr}$ is the patient list corresponding to the master solution: $\hat{\mathcal{P}}_{hdr} = \{p \in \mathcal{P} \mid \hat{x}_{hdpr} = 1\}$. This cut provides a lower bound (LB) for Q^s_{hdr}: the cancellation cost corresponding to patients in $\hat{\mathcal{P}}_{hdr}$ is $\bar{Q}^s_{hdr}$, and if we remove a patient $p \in \hat{\mathcal{P}}_{hdr}$ from the current assignment, we can potentially save the corresponding cancellation cost, c^{cancel}_p.

Note that the LBBD cut is tight at the current master solution, i.e., it provides the exact cancellation cost for the master solution $(\hat{u}_{hd}, \hat{y}_{hdr}, \hat{x}_{hdpr}, \hat{w}_p)$. More importantly, different than the vanilla no-good cut, it also provides a (not necessarily tight) LB for any assignment with a patient list that is either a superset of $\hat{\mathcal{P}}_{hdr}$, or contains a subset of $\hat{\mathcal{P}}_{hdr}$.

Theorem 1 states that the proposed LBBD cut is valid, which means it eliminates the current master solution, and more importantly it does not exclude any optimal solution of (DE). Considering that there are finitely many binary x_{hdpr} feasible solutions in the master problem, this result implies that the LBBD algorithm with the cuts (6) converge to an optimal solution of (DE) in a finite number of iterations.

Theorem 1. *The LBBD optimality cut (6) is valid.*

The proof for the validity of the LBBD cuts is given in the Online Supplement.

5.3 BDD-based Benders Cuts

The next category of cutting planes, which we call *BDD-based Benders cuts*, are based on the work of Lozano and Smith (2018). Those cuts are applicable here because (i) in our decomposition the master variables that are linked to the subproblems, i.e. the x-variables, are all binary; and (ii) any subproblem constraint that is impacted by the first-stage decisions is deactivated by a *single* x_{hdpr} variable. Despite the fact that the subproblems are BIPs, the BDD-based Benders cuts are actually a set of classical Benders cuts; they are obtained by first transforming a subproblem into a shortest path problem, then generating classical Benders optimality cuts from the reformulation which is now an LP. In what follows, we first provide some basic concepts for BDDs, and then explain how to transform our subproblems into shortest path problems via BDDs and in turn obtain the additional set of Benders cuts.

A BDD is a layered directed acyclic graph with a single root node and a single terminal node, which is used to represent a BIP, e.g., see the example in Figure 1. The arcs of a BDD correspond to assigning values to binary variables. More specifically, each layer of arcs corresponds to a binary variable. There are two types of arcs, namely zero-arc and one-arc, respectively corresponding to the associated variable taking the value of zero or one. There is a one-to-one correspondence between feasible solutions of the BIP and root-to-terminal paths in the BDD, thus the BDD compactly represents the feasible set of the BIP. Moreover, arcs in the BDD are assigned length values in such a way that for each root-to-terminal path, its path length is equal to the BIP objective function

value of the corresponding BIP feasible solution. Therefore, finding an optimal solution to BIP with a minimization/maximization objective reduces to finding a shortest/longest path in the BDD. For more details on BDDs, we refer the reader to the book by Bergman et al. (2016).

In our problem, we transfer each subproblem (5) to a BDD. We note that although the BDD size can be exponential in the BIP size, it is pseudo-polynomial in our case since our subproblem is a knapsack problem. We first provide a simple example to illustrate the BDD transformation.

Example 1. Suppose in the subproblem (5) for an (h, d, r, s) tuple there are four scheduled patients ($p = 1, ..., 4$), and we need to decide whether to cancel them. Assume that for those four patients their cancellation costs are respectively $4, 1, 3, 8$ and their surgery durations in the current scenario are respectively $2, 1, 3, 3$, then the subproblem formulation is the following (we have replaced all $\hat{x}_{hdpr}, p = 1, ..., 4$ in (5) with 1 as all four patients are selected):

$$\bar{Q}^s_{hdr} = \min\ 4(1 - z^s_{hd1r}) + (1 - z^s_{hd2r}) + 3(1 - z^s_{hd3r}) + 8(1 - z^s_{hd4r}) \tag{7a}$$

$$\text{s.t. } 2z^s_{hd1r} + z^s_{hd2r} + 3z^s_{hd3r} + 3z^s_{hd4r} \leq 5 \tag{7b}$$

$$z^s_{hdpr} \leq 1 \qquad \forall p \in \hat{\mathcal{P}}_{hdr} \tag{7c}$$

$$z^s_{hdpr} \in \{0, 1\} \qquad \forall p \in \hat{\mathcal{P}}_{hdr} \tag{7d}$$

where $\hat{\mathcal{P}}_{hdr} = \{1, 2, 3, 4\}$ represents the subset of selected patients. To simplify the notation, we equivalently rewrite the objective as:

$$\bar{Q}^s_{hdr} - 16 = \min \quad -4z^s_{hd1r} - z^s_{hd2r} - 3z^s_{hd3r} - 8z^s_{hd4r} \tag{8}$$

The problem is represented via the BDD in Figure 1. The BDD contains a root node at the

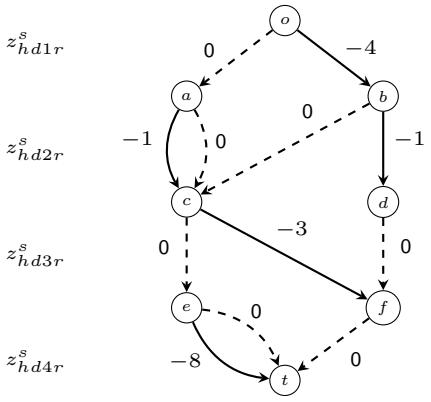

Figure 1: BDD for a simple subproblem with four patients

top, denoted by o, and a terminal node at the bottom, denoted by t. Each layer of the arcs represents the values of the variables z^s_{hdpr}, from patient 1 to patient 4. (Note that different variable orderings can produce different BDDs, thus may impact the computational performance.) At each layer, the dashed and solid arcs respectively indicate selecting 0 and 1 values for the layer's associated variable. The value above or beside each arc is the cost of z^s_{hdpr} if it receives the value

indicated by the arc type. When transforming the subproblem to a shortest path problem, those values are used as costs for the arcs. Each path from the root node o to the terminal node t represents a feasible solution for the subproblem, and each feasible solution is represented by a path from o to t. Also, the path length provides the correct evaluation of the objective value for the corresponding feasible solution. For example, the path *o-b-d-f-t* represents the feasible solution $z_{hd1r} = 1, z_{hd2r} = 1, z_{hd3r} = 0, z_{hd4r} = 0$. The length of this path is -5, which equals the objective value of the corresponding z solution. Therefore, by finding a shortest path from the root node to the terminal node, we can obtain an optimal solution to the subproblem. □

We now formally illustrate the BDD transformation of the subproblem (5). The following discussion deals with the subproblem corresponding to an (h, d, r, s) tuple, and makes cancellation decisions concerning the scheduled patient set $\hat{\mathcal{P}}_{hdr}$. We only consider patients from $\hat{\mathcal{P}}_{hdr}$ in the BDD reformulation. Let $\mathcal{G}^s_{hdr} = \{\mathcal{N}^s_{hdr}, \mathcal{A}^s_{hdr}\}$ be the BDD for this subproblem where $\mathcal{N}^s_{hdr}$ is the set of nodes and $\mathcal{A}^s_{hdr}$ is the set of arcs. Let $\mathcal{A}^s_{hdr0}$ and $\mathcal{A}^s_{hdr1}$ be the sets of zero-arcs and one-arcs, i.e., corresponding to setting variables in $\{z^s_{hdpr}\}_{p\in\hat{\mathcal{P}}_{hdr}}$ equal to 0 and 1 in the subproblem, respectively. In addition, we introduce the sets $\mathcal{A}^s_{hdpr0} \subset \mathcal{A}^s_{hdr0}$ and $\mathcal{A}^s_{hdpr1} \subset \mathcal{A}^s_{hdr1}$ to respectively denote the sets of zero-arcs and one-arcs corresponding to a patient p. Nodes $o \in \mathcal{N}^s_{hdr}$ and $t \in \mathcal{N}^s_{hdr}$ are the root and terminal nodes of the diagram $\mathcal{G}^s_{hdr}$. The capacity of all arcs are one. We denote the cost of an arc $a \in \mathcal{A}^s_{hdr}$ as g^s_{hdra} and its value is decided as follows:

$$g^s_{hdra} = \begin{cases} -c^{\text{cancel}}_p, & \text{if } a \in \mathcal{A}^s_{hdpr1} \\ 0, & \text{if } a \in \mathcal{A}^s_{hdr0} \end{cases}$$

Let the start and end points of arc a be $s(a)$ and $d(a)$, respectively. Then the BDD reformulation of the subproblem as a capacitated shortest path problem is as follows:

$$\bar{Q}^s_{hdr} - \sum_{p\in\mathcal{P}} c^{\text{cancel}}_p \hat{x}_{hdpr} = \min \sum_{a\in\mathcal{A}^s_{hdr}} g^s_{hdra} f_a \tag{9a}$$

$$\text{s.t.} \sum_{a\in\mathcal{A}^s_{hdr}|s(a)=o} f_a = 1 \qquad (\pi_o) \tag{9b}$$

$$\sum_{a\in\mathcal{A}^s_{hdr}|s(a)=i} f_a - \sum_{a\in\mathcal{A}^s_{hdr}|d(a)=i} f_a = 0 \ \forall i \in \mathcal{N}^s \setminus \{o, t\} \qquad (\pi_i) \tag{9c}$$

$$\sum_{a\in\mathcal{A}^s_{hdr}|d(a)=t} f_a = 1 \qquad (\pi_t) \tag{9d}$$

$$f_a \le \hat{x}_{hdpr} \qquad \forall a \in \mathcal{A}^s_{hdpr1}, p \in \mathcal{P} \quad (-\xi_a) \tag{9e}$$

$$f_a \ge 0 \qquad \forall a \in \mathcal{A}^s_{hdr} \tag{9f}$$

where f_a is the variable for the flow through arc a. The variable in parentheses at the end of each constraint is the corresponding LP dual variable. Note that the objective value of problem (9) is different from that of the subproblem (5) by a constant term $\sum_{p\in\mathcal{P}} c^{\text{cancel}}_p \hat{x}_{hdpr}$. This is because in the process of the BDD transformation, we remove the constant term in (5a), $\sum_{p\in\mathcal{P}} c^{\text{cancel}}_p \hat{x}_{hdpr}$, just as shown in the small example, where the objective (7a) becomes (8).

Let $\bar{\pi}_o$, $\bar{\pi}_i$ and $\bar{\pi}_t$ be the respective optimal dual values corresponding to constraints (9b), (9c) and (9d). Also define the optimal value for the dual variable ξ_a as $\bar{\xi}_a$. We can derive the following classical Benders cut from the problem (9), which we refer to as *BDD-based Benders cut*:

$$Q^s_{hdr} - \sum_{p\in\mathcal{P}} c^{\text{cancel}}_p x_{hdpr} \geq \bar{\pi}_o - \sum_{p\in\mathcal{P}} \Big(\sum_{a\in\mathcal{A}^s_{hdpr1}} \bar{\xi}_a \Big) x_{hdpr} \tag{10}$$

Note that $\bar{\pi}_t$ is set to 0 and thus not included in the BDD-based Benders cut. We are able to do this because the corresponding constraint of π_t, (9d), is linearly dependent on the other flow balance constraints (9b)-(9c).

As suggested by Lozano and Smith (2018), this cut can be further strengthened by replacing the x_{hdpr} coefficient of $\sum_{a\in\mathcal{A}^s_{hdpr1}} \bar{\xi}_a$, with $\max_{a\in\mathcal{A}^s_{hdpr1}} \bar{\xi}_a$:

$$Q^s_{hdr} - \sum_{p\in\mathcal{P}} c^{\text{cancel}}_p x_{hdpr} \geq \bar{\pi}_o - \sum_{p\in\mathcal{P}} \Big(\max_{a\in\mathcal{A}^s_{hdpr1}} \bar{\xi}_a \Big) x_{hdpr} \tag{11}$$

In our implementation we use this strengthened version of BDD-based Benders cut. For the proof of the validity of (11), we refer the readers to Lozano and Smith (2018).

5.4 Classical Benders Cuts from the Relaxed Subproblem

In order to provide a better LB for Q^s_{hdr}, we also add classical Benders cuts from the LP relaxation of the subproblem (5) to the master problem. Note that unlike the LBBD cuts and the BDD-based Benders cuts, the classical Benders cuts generated in this method cannot guarantee convergence. However, it speeds up the convergence in our algorithm when used together with LBBD cuts or BDD-based Benders cuts.

The process to obtain classical Benders cuts is standard. We linearly relax the subproblem (5) and denote the optimal objective value of the relaxed problem as $\bar{Q}^{s\text{LP}}_{hdr}$. Let $\bar{\delta}$ and $\bar{\eta}$ denote the optimal dual solutions that respectively correspond to constraints (5b) and (5c) in the LP relaxation. The classical Benders cut (12), which is generated and added to the master problem (4) when $\hat{Q}^s_{hdr} < \bar{Q}^{s\text{LP}}_{hdr}$, is as follows:

$$Q^s_{hdr} \geq \sum_{p\in\hat{\mathcal{P}}_{hdr}} (c^{\text{cancel}}_p + \bar{\delta}_p) x_{hdpr} + \bar{\eta} B_{hd} \tag{12}$$

5.5 Additional Algorithmic Enhancements

In order to further improve the performance of the algorithm, we introduce three enhancement techniques. Section 5.5.1 provides a heuristic way to find an initial solution, thus a good upper bound (UB) to start the search. Section 5.5.2 derives a relaxation of the subproblem which is used to tighten the master formulation. Section 5.5.3 introduces a scheme that inserts additional heuristic solutions in the branch-and-cut algorithm to improve UBs.

5.5.1 Adapted First Fit Decreasing (FFD) Heuristic.

FFD is a heuristic method that finds a good feasible solution for the binary bin-packing problem (Johnson et al., 1974). We choose to use this heuristic because it is quick to implement, and is flexible to adapt to our problem setup. More importantly, FFD used together with the LBBD method achieves good results in the literature of scheduling and assignment problems (Fazel-Zarandi et al., 2013; Roshanaei et al., 2017). We propose an adapted version of FFD to find an initial assignment of the patients for the SDORS problem. Viewing the ORs as bins and their operating time limits as the bin capacities, and the surgery duration of a patient as the weight of an item that needs to be fitted, the goal is to schedule as many patients to the ORs as possible.

To find an initial assignment via the heuristic, we first sort the (h, d) pairs in the decreasing order of the operating time limit B_{hd}, then sort the ORs from different (h, d) pairs according to the sorted order of (h, d) pairs. For ORs of the same (h, d) pair, arrange them in the increasing order of their indices. Note that because we have both mandatory and non-mandatory patients, the FFD algorithm is adapted to accommodate the requirement that mandatory patients must be arranged in the current planning horizon. More specifically, when we initially sort the patients, we first sort all the mandatory patients, i.e. before sorting any non-mandatory patients. Both the mandatory and non-mandatory patients are sorted in the decreasing order of their health status score ω_p.

Overall, the heuristic has the following steps: (i) Randomly pick a scenario $s \in \mathcal{S}$ (in the implementation we pick the first scenario). (ii) For each patient in the sorted order, check whether one of the opened ORs has enough capacity left to fit the patient based on the surgery duration in the selected scenario, starting from the first OR in the sorted order. If one of the ORs has enough capacity, assign the patient to the OR and update the capacity of the OR. If not, open a new OR to fit the patient in. (iii) If there are no new OR left to fit the patient, postpone the patient if she is not mandatory, otherwise the FFD heuristic cannot schedule all mandatory patients, and other heuristics could be tried. (In our experiments, we get feasible solutions from FFD for all instances, i.e., never encountered any instance where FFD cannot schedule mandatory patients.) (iv) After assigning all the patients to ORs, we solve the subproblem (5) for each opened ORs to find its corresponding cancellation cost under this assignment.

We use this heuristic solution as a warm start which provides a good initial UB for the algorithm. In addition, based on the heuristic solution, we derive LBBD cuts (6) or BDD-based Benders cuts (11), plus classical Benders cuts (12). We add those cuts to the master problem (4) before the branch-and-cut algorithm, which we will describe in more detail in Section 5.6. Adding those cuts helps to reduce the initial optimality gap of the decomposition algorithm.

5.5.2 Adding Subproblem Relaxations to the Master Problem.

In decomposition algorithms, it is observed that adding some form of relaxed subproblem to the master problem may greatly improve computational efficiency (Ciré et al., 2016). In our algorithm, we tighten the master problem by providing a LB for each Q^s_{hdr}, which can be derived by relaxing

the subproblem:

$$Q^s_{hdr} \geq \left(\min_{p \in \mathcal{P}} \frac{c^{\text{cancel}}_p}{T^s_p} \right) \left(\sum_{p \in \mathcal{P}} T^s_p x_{hdpr} - B_{hd} \right) \quad \forall h \in \mathcal{H}, d \in \mathcal{D}, r \in \mathcal{R}_h, s \in \mathcal{S} \tag{13}$$

Intuitively, those constraints approximate the OR cancellation cost by estimating the total overtime and the cost of cancelling overtime patients. We formally state the validity of those constraints in Theorem 2 whose proof is provided in the Online Supplement.

Theorem 2. *Constraints* (13) *provide valid LBs for* Q^s_{hdr}.

5.5.3 Insert Heuristic Solution in Branch-and-Cut.

While implementing the branch-and-cut algorithm, at each integral node we obtain integral solutions for the master variables. However, if the subproblem objectives are underestimated at those nodes, those master variable solutions are not utilized to obtain a UB. Inspired by the practice, e.g., in Bodur and Luedtke (2016), we combine the master variable solution at an integral node and the correct value for the corresponding subproblem, and provide this additional feasible solution to the solver. More specifically, each additional solution is obtained at a master solution, $(\hat{u}, \hat{y}, \hat{x}, \hat{w}, \hat{Q})$, when $(\hat{u}, \hat{y}, \hat{x}, \hat{w})$ is *integral*. We solve the corresponding subproblems for this master solution and get $\bar{Q}$, then the solution $(\hat{u}, \hat{y}, \hat{x}, \hat{w}, \bar{Q})$ is a feasible solution for (DE). Next, we evaluate the objective value at $(\hat{u}, \hat{y}, \hat{x}, \hat{w}, \bar{Q})$. If it is better than the incumbent UB of branch-and-cut algorithm at the moment, then we update the incumbent UB by setting $(\hat{u}, \hat{y}, \hat{x}, \hat{w}, \bar{Q})$ as a heuristic solution via the commercial solver callback. We add additional solutions through the commercial solver callback inside the branch-and-cut algorithm.

5.6 Overall Implementation Approach

As mentioned before, we add the cutting planes, including the LBBD cuts (6), the BDD-based Benders cuts (11), and the classical Benders cuts (12), through branch-and-cut. This is different from the standard cutting plane implementation, where master problem need to be solved again each time when new cutting planes are generated.

The overall implementation of the two-stage decomposition algorithm is illustrated as a flow chart in the Online Supplement. In the rest of this section we describe the overall implementation of our decomposition algorithm. We divide the algorithm into two phases for the ease of explanation. In phase one we use the adapted FFD and subproblem relaxation to tighten the UB and LB of the master problem, before entering phase two where we run the branch-and-cut algorithm:

Phase one: First, we use the adapted FFD heuristic to obtain an initial solution. This solution is added as a *warm start* in the commercial solver to provide a feasible solution before the start of the branch-and-cut algorithm. We also generate LBBD cuts or BDD-based Benders cuts, as well as classical Benders cuts from this solution and add them to the master problem. Next, we generated the constraints (13) from the subproblem relaxations and also add them to the master problem.

Phase two: We obtain the master problem with extra cuts and constraints from phase one and solve it with branch-and-cut. In the branch-and-cut algorithm at each branch-and-bound node, some integer variables are restricted while the others are linearly relaxed to obtain a node LP relaxation. Solve this LP. If the objective value is greater than or equal to the incumbent UB, then the current node can be pruned. Otherwise we proceed to check the integrality of $(\hat{u}, \hat{y}, \hat{x}, \hat{w})$ in the master solution. If $(\hat{u}, \hat{y}, \hat{x}, \hat{w})$ is integral, then we solve the corresponding subproblems (5) and subproblem LP relaxations, and get their respective optimal objective values $\bar{Q}$ and $\bar{Q}^{\text{LP}}$. We check if the incumbent UB can be updated with a heuristic solution as described in Section 5.5.3. Also, we generate LBBD cuts or BDD-based Benders cuts and classical Benders cuts. In the CPLEX lazy constraint callback, compare the master solution of $\hat{Q}$ with $\bar{Q}$ and $\bar{Q}^{\text{LP}}$. If $\hat{Q} < \bar{Q}$, we add LBBD cuts or BDD-based Benders cuts; if $\hat{Q} < \bar{Q}^{\text{LP}}$, we add classical Benders cuts. On the other hand, if some elements in the solution $(\hat{u}, \hat{y}, \hat{x}, \hat{w})$ are fractional, we only solve the subproblem LP relaxations, obtain $\bar{Q}^{\text{LP}}$, generate the classical Benders cuts and implement them if $\hat{Q} < \bar{Q}^{\text{LP}}$ within the CPLEX user cut callback. We refer to those classical Benders cuts added at fractional solutions as *user cuts*. Notice that in order to avoid adding too many user cuts to the point of slowing down the algorithm, we do the following *user cut management*: we only add user cuts at the root node and every 150 nodes afterwards, and after 4000 nodes are processed in the branch-and-bound tree, we no longer add any user cut. In addition, we add at most 50 user cuts at the root node and at most 5 user cuts at any node afterwards.

After cutting planes are added in the CPLEX lazy constraint callback or the user cut callback, the node LP relaxation is solved again with those additional cutting planes. We repeat this process, until the stopping criteria is met, i.e. the gap between branch-and-bound UB and LB is small enough. In our implementation we stop the algorithm when such a gap is no more than 1%.

6 Three-stage Decomposition

As an alternative to the two-stage decomposition provided in Section 5, we also develop a three-stage decomposition for (DE). Here, we first decompose it into a master problem and a set of subproblems, and develop a class of LBBD optimality cuts to connect them. However, this time each subproblem is more complex, because it is in fact itself a 2SIP. Therefore, we further develop a two-stage decomposition for the subproblems.

In Section 6.1, we introduce the LBBD algorithm for the (DE). In Section 6.2, we show how the computationally expensive LBBD subproblems can be further decomposed via the BDD-based method. Additional algorithmic enhancements, some of which are similar to those in the two-stage decomposition, are also used in the three-stage decomposition, which we cover in Section 6.3.

6.1 Decomposition of the SAA Problem

We first decompose (DE) into two stages that are linked via LBBD cuts. In order to distinguish this decomposition with the one in Section 6.2, we use the conventional names *LBBD master problem*

and *LBBD subproblem* for its master and subproblems, respectively.

6.1.1 LBBD Decomposition Framework.

The LBBD master problem includes the original variables u_{hd} and w_p, the new variables y_{hd}, x_{hdp}, and a new variable Q_{hd}. That is, in the master problem we only decide if patients should be assigned to a hospital on a specific day, leaving the patient to room assignment to the subproblems. Thus, we drop the indices $r \in \mathcal{R}_h$ in the original variables, and get new variables y_{hd} and x_{hdp}. Note that y_{hd} is a nonnegative integer variable, representing the number of ORs opened in hospital h on day d. Also we use the variable Q_{hd} to denote the *expected* cancellation cost for an (h, d) pair:

$$\min \quad \sum_{h\in\mathcal{H}}\sum_{d\in\mathcal{D}} G_{hd}u_{hd} + \sum_{h\in\mathcal{H}}\sum_{d\in\mathcal{D}} F_{hd}y_{hd} + \sum_{h\in\mathcal{H}}\sum_{d\in\mathcal{D}}\sum_{p\in\mathcal{P}} c_{dp}^{\text{sched}}x_{hdp} + \sum_{p\in\mathcal{P}\setminus\mathcal{P}'} c_p^{\text{unsched}}w_p + \sum_{h\in\mathcal{H}}\sum_{d\in\mathcal{D}} Q_{hd} \tag{14a}$$

$$\text{s.t.} \quad \sum_{h\in\mathcal{H}}\sum_{d\in\mathcal{D}} x_{hdp} = 1 \qquad \forall p \in \mathcal{P}' \tag{14b}$$

$$\sum_{h\in\mathcal{H}}\sum_{d\in\mathcal{D}} x_{hdp} + w_p = 1 \qquad \forall p \in \mathcal{P}\setminus\mathcal{P}' \tag{14c}$$

$$y_{hd} \le |\mathcal{R}_h|u_{hd} \qquad \forall h \in \mathcal{H}, d \in \mathcal{D} \tag{14d}$$

$$y_{hd} \ge x_{hdp} \qquad \forall h \in \mathcal{H}, d \in \mathcal{D}, p \in \mathcal{P} \tag{14e}$$

$$\text{[LBBD cuts]} \tag{14f}$$

$$u_{hd}, x_{hdp} \in \{0,1\} \qquad \forall h \in \mathcal{H}, d \in \mathcal{D}, p \in \mathcal{P}, r \in \mathcal{R}_h \tag{14g}$$

$$w_p \in \{0,1\} \qquad \forall p \in \mathcal{P} \tag{14h}$$

$$y_{hd} \in \mathbb{Z}^+, \qquad \forall h \in \mathcal{H}, d \in \mathcal{D} \tag{14i}$$

$$Q_{hd} \ge 0, \qquad \forall h \in \mathcal{H}, d \in \mathcal{D} \tag{14j}$$

Constraints (14b) and (14c) assign patients to hospitals, as well as to days of the current planning horizon or a future time. Constraints (14d) ensure that when the surgical suite of hospital h is not opened on a day d, there is no OR in that hospital to be opened. Constraints (14e) make sure that when a hospital h has a patient assigned on day d, there must be at least one OR opened in that hospital. Those constraints are not necessary for a correct formulation, but they make the formulation tighter. Constraints (14f) are LBBD cuts that are generated progressively from the LBBD subproblems.

In this framework we have one subproblem per (h, d) pair, minimizing the expected cancellation cost by finding out the best way to assign patients selected by the master problem to the opened ORs. Via solving the master problem, we get the optimal solution $(\hat{u}_{hd}, \hat{y}_{hd}, \hat{x}_{hdp}, \hat{w}_p, \hat{Q}_{hd})$. Then we pass $\hat{x}_{hdp}$ and $\hat{y}_{hd}$ to the subproblem. The LBBD subproblem for (h, d) is as below:

$$\mathcal{Q}_{hd}(\hat{x}_{hd\cdot}, \hat{y}_{hd}, T^s_{\cdot}) = \min \quad \frac{1}{|\mathcal{S}|}\sum_{s\in\mathcal{S}}\sum_{p\in\mathcal{P}}\sum_{r\in\mathcal{R}_h} c_p^{\text{cancel}}(x_{pr} - z^s_{pr}) \tag{15a}$$

$$\text{s.t.} \quad \sum_{r\in\mathcal{R}_h} x_{pr} = \hat{x}_{hdp} \qquad \forall p \in \mathcal{P} \tag{15b}$$

$$\sum_{p\in\mathcal{P}} T^s_p z^s_{pr} \le B_{hd} \qquad \forall r \in \mathcal{R}_h, s \in \mathcal{S} \tag{15c}$$

$$z^s_{pr} \le x_{pr} \qquad \forall p \in \mathcal{P}, r \in \mathcal{R}_h, s \in \mathcal{S} \tag{15d}$$

$$x_{pr} = 0 \qquad \forall r \ge \hat{y}_{hd} + 1 \tag{15e}$$

$$x_{pr}, z^s_{pr} \in \{0,1\} \qquad \forall p \in \mathcal{P}, r \in \mathcal{R}_h, s \in \mathcal{S} \tag{15f}$$

Constraints (15b) require that the patient to OR assignment for the current (h,d) pair should be consistent with the assignment decision in the master problem. Constraints (15c) specify time limit. Constraints (15d) ensure that if a patient is not scheduled then she will not be operated on. Constraints (15e) are symmetry breaking constraints for ORs. Those constraints ensure that for a hospital-day pair, ORs with lower indices are opened before ORs with higher indices. These are valid because ORs in the same hospital are homogeneous. Also note that those constraints are connected to the master problem by $\hat{y}_{hd}$, which means at most $\hat{y}_{hd}$ many ORs can be used. Moreover, as the master tries to minimize opened ORs due to opening costs, at the optimal solution, the subproblems will use the exact number of ORs suggested by the master.

6.1.2 LBBD cuts.

The LBBD subproblem is always feasible, therefore there is no need to introduce LBBD feasibility cuts. Using $\bar{Q}_{hd}$ to denote $\mathcal{Q}_{hd}(\hat{x}_{hd\cdot}, \hat{y}_{hd}, T^s_\cdot)$, when $\hat{Q}_{hd} < \bar{Q}_{hd}$, we add the followings cuts, which we refer to as the LBBD optimality cut, to the LBBD master problem:

$$Q_{hd} \ge \bar{Q}_{hd} \left(g_{hdj} - \sum_{p\in\hat{\mathcal{P}}_{hd}} (1 - x_{hdp}) \right) \tag{16a}$$

$$y_{hd} \ge (1 + \hat{y}_{hd})(1 - g_{hdj}) \tag{16b}$$

$$g_{hdj} \in \{0,1\} \tag{16c}$$

where $\hat{\mathcal{P}}_{hd} = \{p \in \mathcal{P} \mid \hat{x}_{hdp} = 1\}$ is the current patient list, index $j = \hat{y}_{hd}$, and g_{hdj} is a binary variable that equals 1 when $y_{hd} \le \hat{y}_{hd}$, otherwise 0.

This cut is deducted from the logic $y_{hd} \le \hat{y}_{hd} => Q_{hd} \ge \bar{Q}_{hd} - \sum_{p\in\hat{\mathcal{P}}_{hd}} \bar{Q}_{hd}(1 - x_{hdp})$, which means the LB on Q_{hd} is imposed only when $y_{hd} \le \hat{y}_{hd}$. The lower-bounding inequality ensures that when the current patient list or a superset of it occurs, the corresponding cancellation cost will not be lower than $\bar{Q}_{hd}$. We state the validity of the LBBD cut in Theorem 3, of which the proof is provided in the Online Supplement.

Theorem 3. *The LBBD optimality cuts (16) are valid.*

A nice feature of cuts (16) is that at most $\sum_{h\in\mathcal{H}} \sum_{d\in\mathcal{D}} (|\mathcal{R}_h|+1)$ many binary variables g_{hdj} are needed, *regardless* of the number of LBBD cuts that are eventually generated. Therefore, in our implementation we create all potential variables $\{g_{hdj} | h \in \mathcal{H}, d \in \mathcal{D}, j = 0, 1, ..., |\mathcal{R}_h|\}$ before the

algorithm starts. This is important because the solver, CPLEX, does not allow the addition of new variables during the branch-and-bound process. This also means we do not need to add too many new variables in order to use these cuts.

6.2 Decomposition of the LBBD Subproblems

The LBBD subproblem (15) is itself a 2SIP, whose first stage decision is to assign the patients in the list $\hat{\mathcal{P}}_{hd}$ to ORs, which corresponds to the decision variables x_{pr}. The recourse decision concerns whether to operate a surgery under each scenario, represented by the variables z^s_{pr}. Solving such a 2SIP can be time consuming. For example in our experiment, for an instance with $|\mathcal{S}|= 50, |\mathcal{P}|= 20, |\mathcal{H}|= 3, |\mathcal{R}_h|= 3$, when the LBBD master problem assigns all patients to a single (h, d) pair, the subproblem 2SIP could not be solved to optimality by the commercial solver after one hour.

Because the subproblem is a 2SIP, we are able to use BDD-based Benders cuts in the decomposition, following a similar approach as in Section 5.3. To distinguish from the LBBD decomposition in the previous section, we name the master problem and a subproblem in this decomposition as the *BDD master problem* and *BDD subproblem*, respectively.

The BDD master problem assigns patients to ORs in an (h, d) pair, and relaxes the constraints on the OR operating time limit. The continuous variabls θ_{sr} corresponds to the optimal objective value of the subproblem for OR r in scenario s. The BDD master problem is as follows:

$$\min \quad \sum_{p\in\mathcal{P}}\sum_{r\in\mathcal{R}_h} c^{\text{cancel}}_p x_{pr} + \frac{1}{|\mathcal{S}|}\sum_{s\in\mathcal{S}}\sum_{r\in\mathcal{R}_h}\theta_{sr} \tag{17a}$$

$$\text{s.t.} \quad \sum_{r\in\mathcal{R}_h} x_{pr} = \hat{x}_{hdp} \qquad \forall p \in \mathcal{P} \tag{17b}$$

$$x_{pr} = 0 \qquad \forall r \geq \hat{y}_{hd} + 1 \tag{17c}$$

$$-\sum_{p\in\mathcal{P}} c^{\text{cancel}}_p x_{pr} \leq \theta_{sr} \leq 0 \qquad \forall s \in \mathcal{S}, r \in \mathcal{R}_h \tag{17d}$$

$$\text{[BDD-based Benders cuts]} \tag{17e}$$

$$x_{pr} \in \{0, 1\} \qquad \forall p \in \mathcal{P} \tag{17f}$$

The left-hand side of the equation (17d) is derived from the relaxation of BDD subproblem (18), by relaxing its time limit constraint. Constraints (17e) are BDD-based Benders cuts generated from the shortest path problem on a BDD transformed from subproblem (18). All the other constraints have the same meaning as their counterparts in LBBD subproblem (15).

We use $\check{x}_{pr}$ to denote the optimal BDD master solution for x_{pr} and pass it to the BDD subproblem. We observe that the subproblem decomposes with scenarios and ORs, yielding the BDD subproblem for each scenario s and OR r as:

$$\theta_{sr}(\check{x}_{\cdot r}, T^s_{\cdot}) = \min \quad \sum_{p\in\mathcal{P}} -c^{\text{cancel}}_p z^s_{pr} \tag{18a}$$

$$\text{s.t.} \quad \sum_{p\in\mathcal{P}} T^s_p z^s_{pr} \leq B_{hd} \tag{18b}$$

$$z^s_{pr} \leq \check{x}_{pr} \qquad \forall p \in \mathcal{P} \tag{18c}$$
$$z^s_{pr} \in \{0,1\} \qquad \forall p \in \mathcal{P} \tag{18d}$$

Constraints (18b) and (18c) are respectively the time limit constraints and the constraints for UB of z^s_{pr}. Like before, we simplify the notations and use $\ddot{\theta}_{sr}$ to denote $\theta_{sr}(\check{x}_{\cdot r}, T^s_{\cdot})$.

Observe that this problem is almost the same as problem (5), except for the constant term in the objective and hospital-day indices. Therefore, we can derive strengthened BDD-based Benders cuts similar to (11).

6.3 Additional Algorithmic Enhancements for the Three-stage Decomposition

In this section we introduce several algorithmic enhancements for the three-stage decomposition. In Section 6.3.1 we derive classical Benders cuts from LBBD and BDD subproblems. Section 6.3.2 introduces a scheme that avoids solving some LBBD subproblems to optimality by identifying suboptimal master solutions early on. Sections 6.3.3 and 6.3.4 show how to implement an adapted FFD heuristic and add the relaxed subproblem to the master problem in the three-stage decomposition. Section 6.3.5 explains the overall implementation of the three-stage decomposition algorithm with enhancements. Note that in the three-stage decomposition we also provide additional heuristic solutions in the branch-and-cut implementation the same way as described in Section 5.5.3, which we will also discuss in Section 6.3.5.

6.3.1 Classical Benders Cuts.

For both the LBBD subproblem and the BDD subproblem, we can derive classical Benders cuts from their LP relaxations, respectively providing LBs for Q_{hd} and θ_{sr}. The procedure of deriving those classical Benders cuts is standard. We describe the process and the classical Benders cuts in the Online Supplement.

6.3.2 Early Stopping Scheme in the LBBD Branch-and-cut Implementation.

We apply an early stopping scheme (Karwan, 1976) which identifies a suboptimal LBBD master problem solution early on, so we can avoid solving the corresponding 2SIP LBBD subproblems to optimality. This is useful as the LBBD subproblems can be very time consuming to solve, especially considering the LBBD master problem assigns patients to (h, d) pairs without knowledge about the operating time limit of ORs. Such lack of knowledge sometimes leads to the LBBD master problem generating suboptimal solutions that over-schedule patients in some (h, d) pairs and results in high cancellation costs. We would like to identify those suboptimal solutions as soon as possible, because when the LBBD master problem passes a heavily over-scheduled patient list to an (h, d) pair, it can be time-consuming to solve the corresponding subproblem.

To understand how the early stopping scheme works, first note that we embed the LBBD cuts into the branch-and-bound tree, i.e. in a branch-and-cut framework. While exploring the branch-and-bound tree of the LBBD master problem, we solve an LBBD subproblem at every integral

node. For each integral solution, we record the current best UB of the LBBD master problem as *globalUB* and the operational cost at the current integral solution as *incumbentOptCost*. When solving the LBBD subproblem at such a node, we record the best LB of the subproblem whenever one is found, as $\bar{Q}^{\text{LB}}_{hd}$. If *globalUB* < *incumbentOptCost*+$\bar{Q}^{\text{LB}}_{hd}$, there is no need to continue solving the subproblem, as the current LBBD master solution cannot be optimal.

In short, we stop solving an LBBD subproblem once we detect that its lower bound is high enough, and will result in a higher master problem objective value than the best-known UB of the master problem in the branch-and-bound tree.

Although when the subproblem is stopped early it is not solved to optimality, it is still helpful to derive an LBBD cut from its best-known LB, $\bar{Q}^{\text{LB}}_{hd}$. Such a cut is useful to cut off the current master solution, and has the same logic as (16), though we need to replace $\bar{Q}_{hd}$ in (16) with $\bar{Q}^{\text{LB}}_{hd}$:

$$Q_{hd} \geq \bar{Q}^{\text{LB}}_{hd} \left(g_{hdj} - \sum_{p \in \hat{\mathcal{P}}_{hd}} (1 - x_{hdp}) \right) \tag{19a}$$

$$y_{hd} \geq (1 + \hat{y}_{hd})(1 - g_{hdj}) \tag{19b}$$

$$g_{hdj} \in \{0, 1\} \tag{19c}$$

which tells the LBBD master problem that for the current (h, d) pair, if we open no more than $\hat{y}_{hd}$ rooms and assign all the patients in $\hat{\mathcal{P}}_{hd}$ to it, the cancellation cost is at least $\bar{Q}^{\text{LB}}_{hd}$.

6.3.3 Adapted FFD Heuristic.

We use the adapted FFD heuristic to get an initial solution for the LBBD master problem, following the same steps as described in Section 5.5.1. We also develop an adapted FFD heuristic for the BDD master problem to assign patients in $\hat{\mathcal{P}}_{hd}$ to the $\hat{y}_{hd}$ opened ORs. Since the ORs are homogenous in the same (h, d), without loss of generality we select ORs with the smallest $\hat{y}_{hd}$ indices to open, and sort them in the order of their indices. We sort the patients in $\hat{\mathcal{P}}_{hd}$ by the decreasing order of their cancellation costs, and assign patients to ORs in their sorted order. Based on the surgery duration in the first scenario, for each patient we try to fit her into one of the opened ORs. If none of the opened ORs can fit her, we use an OR that has not been opened. If a patient cannot be fitted into any of the ORs and all the ORs have been used, we assign her to the last OR in the sorted order. After assigning all the patients, we solve a BDD subproblem with this assignment to find the corresponding values of $\ddot{\theta}_{sr}$. This heuristic solution is used before the branch-and-cut process as a warm start. We also derive BBD-based Benders cuts and classical Benders cuts based on the heuristic solution and add them to the BBD master problem (17).

6.3.4 Adding Subproblem Relaxations to the Master Problem.

We derive the following constraints from the LBBD subproblems and add them to the LBBD master problem, to provide lower bounds for the variables $\{Q_{hd}\}_{h\in\mathcal{H},d\in\mathcal{D}}$:

$$Q_{hd} \geq \frac{1}{|\mathcal{S}|}\sum_{s\in\mathcal{S}}\left(\min_{p\in\mathcal{P}}\frac{c_p^{\text{cancel}}}{T_p^s}\right)\left(\sum_{p\in\mathcal{P}}T_p^s x_{hdp} - B_{hd}y_{hd}\right) \quad \forall h\in\mathcal{H}, d\in\mathcal{D} \tag{20}$$

and those constraints are valid:

Theorem 4. *The constraints* (20) *are valid for (DE).*

The proof is provided in the Online Supplement.

6.3.5 Overall Implementation.

Similar to the two-stage decomposition, we use a branch-and-cut approach in both the LBBD decomposition and the decomposition of LBBD subproblem. More details on the implementation are provided in the Online Supplement.

7 Computational Results

In this section, we present the computational analysis of the SDORS model and the decomposition algorithms. Section 7.1 provides the information about our data source and the setup of parameters. Section 7.2 includes the results of the SAA analysis, which decides the number of scenarios to use in (DE). Section 7.3 compares the performance of our algorithms with that of the CPLEX solver. Section 7.4 evaluates the value of incorporating stochasticity in the model. Section 7.5 presents results from the sensitivity analysis.

7.1 Parameter Setup

In our analysis we use the same dataset as Roshanaei et al. (2017), which is extracted from the information of 7500 surgeries between 2011 and 2013 from General Surgery Departments of UHN. More specifically, we generate the same parameters B_{hd}, F_{hd}, G_{hd}, c_{dp}^{sched} and c_{dp}^{unsched}, which are presented in the Online Supplement. The only difference in our setup lies in the surgery durations and the cost parameter c_p^{cancel}, as described below.

We assume all surgeries are identical, and surgery durations follow a truncated lognormal distribution (Strum et al., 2000) with mean equals 160 minutes, standard deviation of 40 minutes, and the durations are truncated at 45 minutes and 480 minutes. Those parameters for the surgery duration distribution come from a surgery dataset of UHN.

The health status score of patient p, ω_p, is calculated by $(\alpha_p - |\mathcal{D}|)\rho_p$, where α_p is the number of days the patient has been waiting since her referral date, and ρ_p is the patient's urgency score. The benefit of scheduled patients, c_{dp}^{sched}, the penalty for unscheduled patients, c_{dp}^{unsched}, and the penalty

for cancelled patient, c_p^{cancel}, are calculated as follows: let κ_1 be the unit benefit of scheduling patients on time, κ_2 be the unit penalty of postponing patients to future planning horizons, while κ_3 and κ_4 be the unit penalty of canceling non-mandatory and mandatory patients, respectively. Note that we have $\kappa_1 > 0 > \kappa_2 >> \kappa_3 > \kappa_4$. Then $c_{dp}^{\text{sched}} = \kappa_1 \rho_p (d - \alpha_p)$, $c_p^{\text{unsched}} = \kappa_2 \rho_p (|\mathcal{D}|+1 - \alpha_p)$, $c_p^{\text{cancel}} = \kappa_3 \rho_p (|\mathcal{D}|+1 - \alpha_p)$ for non-mandatory patients, and $c_p^{\text{cancel}} = \kappa_4 \rho_p (|\mathcal{D}|+1 - \alpha_p)$ for mandatory patients. Notice that the benefit of scheduling patients c_{dp}^{sched} depends on the day of scheduling $d \in \mathcal{D}$, which means the earlier a patient is scheduled, the larger this benefit is. $c_{dp}^{\text{sched}} \leq 0$ because it represents the benefit while the model's objective is to minimize the total cost. Since $\alpha \in [60, 120]$, a smaller d will lead to a larger $|c_{dp}^{\text{sched}}|$. We use $|\kappa_1|=$ \$50 and $|\kappa_2|=$ \$5, same as Roshanaei et al. (2017). We cannot directly obtain the cost of cancellation to each patient, so we use the values of κ_1 and κ_2 to estimate these penalties. For the non-mandatory patients' unit cancellation costs, we set $|\kappa_3|=$ \$80 $> |\kappa_1|$, so when a surgery is cancelled, the unit loss will be higher than the benefit of scheduling it. The unit cost of cancelling a mandatory patient is $|\kappa_4|=$ \$100 $> |\kappa_3|$. In Section 7.5 we conduct sensitivity analysis on the cancellation cost.

The details for the values of parameters are provided in the Online Supplement.

7.2 The SAA Analysis

To determine the number of scenarios to use in the (DE), we derive statistically valid LBs and UBs on the optimal value of the original stochastic program, and compare the worst case optimality gap.

Let us rewrite the original 2SIP in a more concise way:

$$\min_{X \in \mathcal{X}} \; f(X) = c^\top X + \mathbb{E}_\xi [Q(X, \xi)] \tag{21}$$

where the set $\mathcal{X}$ include all feasible first-stage variables $X = \{u_{hd}, y_{hdr}, x_{hdpr}, w_p\}$, c represents their objective coefficients, ξ is the vector of random variables that follows a truncated normal distribution, and Q is the second stage value function.

Considering that we use $|\mathcal{S}|$ samples in (DE), in order to obtain a LB on the optimal value of problem (21) via SAA, we first generate N^{Sample} samples where each sample contains $|\mathcal{S}|$ scenarios of simulated surgery durations. Due to the limit of computational resource, N^{Sample} can be relatively small. For each sample n, we solve the (DE) and obtain the optimal solution and objective value. We denote the optimal first-stage solution by X_n and save ~~the~~ it because it will be useful when calculating the UB. We denote the optimal objective value by F_n^{LB}. We can then construct a 95% confidence interval (CI) for the LB of (21) from the N^{Sample} optimal objective values, $\{F_n^{\text{LB}}\}_{n=1,\ldots,N^{\text{Sample}}}$.

To get the UB of SAA, we start with picking a feasible first-stage solution $X \in \mathcal{X}$. The goal is to choose a solution that has the potential to give a better (i.e. lower) objective value. A heuristic for selecting the solution is to choose from the solutions $X_n, n = 1, ..., N^{\text{Sample}}$ as follows: First, generate a medium-sized set of scenarios $\mathcal{S}^{\text{Select}}$. Then for each solution X_n, calculate the second-stage cancelation cost under each scenario $s \in \mathcal{S}^{\text{Select}}$, denoted by $Q(X_n, \xi_s)$. Choose the X_n with

the smallest value of $c^\top X_n + \frac{1}{|\mathcal{S}^{\text{Select}}|}\sum_{s\in\mathcal{S}^{\text{Select}}} Q(X_n, \xi_s)$ and denote it by X^{min}.

After fixing the first-stage solution to $X^{\min}$, we generate a set of scenarios, $\mathcal{S}^{\text{UB}}$. The cardinality of this set should be large. Then for each scenario s we evaluate the cancellation cost and denote it by $Q(X^{\min}, \xi_s)$. Since $X^{\min} \in \mathcal{X}$, the optimal objective value from each scenario, $c^\top X^{\min} + Q(X^{\min}, \xi_s)$, is an UB for the optimal objective value of the corresponding scenario problem. We build a 95% CI from those UBs.

We choose one instance with $|\mathcal{P}|= 12, |\mathcal{H}|= 2, |\mathcal{D}|= 2, |\mathcal{R}_h|= 2$ to conduct the SAA analysis. As for the parameters of the SAA analysis we use $N^{\text{Sample}} = 30$, $|\mathcal{S}^{\text{Select}}|= 1,000$ and $|\mathcal{S}^{\text{UB}}| = 10{,}000$. We compare the worst case optimality gap, obtained as ((Mean of UB + Width of UB) - (Mean of LB - Width of LB)) / (Mean of LB - Width of LB), when $|\mathcal{S}|= 25, 50, 75, 100$. The results are shown in Table 2.

Table 2: SAA optimality gap for different levels of scenarios

	95% CI on LB		95% CI on UB		Worst Case
$\lvert\mathcal{S}\rvert$	Mean	Width	Mean	Width	Opt Gap(%)
25	-127363	239.67	-124145	125.00	2.81
50	-126633	151.93	-124577	109.22	1.83
75	-126175	131.77	-124405	112.17	1.59
100	-125967	133.24	-124510	83.22	1.33

From the results, we can see that when there are 100 scenarios used in the (DE), the optimality gap is below 1.5%. Considering the extensive computational efforts to solve the (DE), we prefer to use a small set of scenarios that can still produce reasonably low optimality gaps. Therefore, we fix the number of scenarios $|\mathcal{S}|= 100$ in the experiments for the following sections.

7.3 Algorithm Comparison

In this section we compare the computational time of the CPLEX solver and our decomposition algorithms. For all the experiments we use C++ API for CPLEX 12.8. We use the default settings of the solver, except that the number of threads is set to one in order to use callbacks for the branch-and-cut algorithm. Note that for the fairness of comparison, we also use one thread when solving directly with CPLEX. The computer used to perform those experiments runs MacOS and has a 2.3 GHz Intel Core i5 processor and a 16GB RAM. The time limit for these experiments is 3 hours and the relative MIP gap is set as 1%.

We generate instances with various sizes: 10 to 75 patients are scheduled to either 2 or 3 hospitals, 3 or 5 days, and 3 or 5 ORs per hospital[1]. The selection for the number of hospitals and days are based on practical reasons, as there are 3 hospitals in the UHN, and the planning horizon of one week contains five business days. We have at most 75 patients and 5 ORs per hospital because further increasing their sizes will result in very large optimality gaps at the end of the time

[1]Those instances, along with the obtained bounds on the optimal objective values, are available for download at https://sites.google.com/site/mervebodr/home/SDORS_Instances.zip?attredirects=0&d=1.

limit. That being said, we believe the instances we sample are diversified enough to cover both sparse and dense cases, and are able to provide an overview for the algorithmic performance.

Table 3: Comparison of Algorithms

instance	Time/Gap				Number of Nodes			
(p-h-d-r)	MIP	2-BDD	2-LBBD	3-LBBD	MIP	2-BDD	2-LBBD	3-LBBD
10-2-3-3	2.8%	1.1%	2.3%	**17.08(min)**	90445	242846	163242	51307
25-2-3-3	8.7%	7.7%	**6.9%**	8.8%	16904	27100	21800	2400
10-3-5-3	4.5%	2.6%	2.9%	**43.68(min)**	47072	177400	160008	276312
25-3-5-3	24.1%	15.5%	16.4%	**9.4%**	30400	23432	27460	7623
50-3-5-3	34.5%	**14.8%**	29.0%	48.0%	3619	6802	5763	0
75-3-5-3	72.1%	**15.5%**	20.2%	46.7%	0	3325	3080	0
10-2-3-5	3.1%	62.06(min)	2.0%	**15.37(min)**	34473	101474	144058	15337
25-2-3-5	7.5%	7.8%	**7.2%**	-	12800	24417	23366	0
50-2-3-5	61.1%	14.8%	**10.6%**	-	7467	9212	10023	0
75-2-3-5	53.3%	**17.3%**	17.5%	-	1	2605	2323	0
10-3-5-5	3.2%	3.0%	4.6%	**18.06(min)**	27090	124901	108930	24361
25-3-5-5	28.5%	**14.4%**	15.5%	-	5000	17894	25506	0
50-3-5-5	59.4%	**18.1%**	20.6%	-	293	7854	8415	0
75-3-5-5	52.7%	22.7%	**16.3%**	-	9	3506	3827	0

The experimental results are shown in Table 3. The instances are denoted by the number of patients (p), hospitals (h), days (d), and rooms (r), linked by dashes. We solve the (DE) directly with CPLEX ("MIP"), as well as solving it using the two-stage decomposition with either only BDD-based cuts ("2-BDD") or only LBBD cuts ("2-LBBD"), and solving it with the three-stage decomposition ("3-LBBD"). We report either the solution time or the optimality gap at the end of the time limit, along with the number of nodes processed in the branch-and-bound tree. In the columns for "Time/Gap" comparison, the bold items represent either the smallest solution time or optimality gap, and the "-" signs represent the cases with over 100% gap. The results in the "Number of Nodes" sections show the number of branch-and-bound nodes processed before the algorithms stop.

The columns of Time/Gap comparisons in Table 3 show that our decomposition algorithms are more efficient than the MIP solver in all test instances. In particular, for the instances with more patients where directly using the MIP solver results in gaps beyond 50%, our decomposition algorithms can solve those instances to a gap below 20%. The three-stage decomposition outperforms the other methods when there are fewer patients to be scheduled. In fact, it performs so well in small instances that in the experiments of Section 7.2 we used this method to solve all the optimization problems to optimality, since the SAA analysis uses a small instance with $|\mathcal{P}|= 12$. For instances with 25 or more patients, the two-stage decomposition methods are the best. Between the two two-stage decompositions, one outperforms the other in almost the same number of tested instances. For the larger instances, the three-stage decomposition becomes very ineffective, as the size of the LBBD subproblems become the bottleneck.

We also note that for the two-stage decomposition, it is also possible to use both BDD-based cuts and LBBD cuts at the same time. However, in most of the instances, adding both cuts does not perform as well as the best performing algorithm that has only one of those two types of cuts. As a potential reason of 2-BDD and 2-LBBD collectively outperforming adding both cuts, we observe that adding extra cuts in each iteration quickly increases the number of cuts in the branch-and-cut algorithm, which slows down the algorithm in later rounds, and overrides the benefit of having an extra type of cut. We do acknowledge though that a careful cut management strategy can potentially make adding both cuts a better option.

In addition, we present results for the objective values of the best integer solutions in the Online Supplement. The results show that if an algorithm has the best computational performance, it is also most likely to produce the best integer solutions.

7.4 Value of Incorporating Stochasticity

We evaluate the value of incorporating stochastic surgery durations in the distributed OR scheduling problem by comparing the optimal schedule from the deterministic DORS model (Roshanaei et al., 2017) and the ones from our stochastic problem (1). In this experiment, we use those instances with three hospitals and five days from Section 7.3, as those instances are the more realistic ones in terms of the number of hospitals and days.

For the evaluation, we first solve both models with a time limit of 3 hours and the optimality gap being set to 1%. Note that after 3 hours, all the deterministic models are solved to optimality, while for the stochastic models we use the results from the fastest algorithms as indicated by Table 3, and most instances are not solved to optimality. After we obtain the hospital and OR opening decisions and patient assignments from the optimization, we evaluate those solutions with randomly generated surgery durations. We randomly generate 10,000 samples of durations, solve the recourse problem (2) with the first stage decisions fixed to the values from the optimization, and calculate the *cancellation rate* and *OR utilization rate* for each sample under different assignments. The cancellation rate equals the ratio between the number of cancelled patients and the total number of scheduled patients. The OR utilization rate is the ratio between the total duration of accepted surgeries and the total available time of all ORs. Finally we obtain the 95% confidence intervals for the evaluations from those 10,000 samples.

In Table 4 we report 95% confidence intervals of the cancellation rate and the OR utilization rate. For each instance, the model with lower cancellation rate and higher OR utilization rate is highlighted. From the table we can conclude that using stochastic model reduces the patient cancellation rate and improves the utilization of ORs. In particular, the improvement is significant for instances with more patients and ORs, e.g., the instances 50-3-5-5 and 75-3-5-5. This can be explained by the fact that the deterministic model opens an average of 28% fewer rooms than the stochastic model, while scheduling about the same number of patients.

It is worth mentioning that the utilization rate we report is calculated considering all available ORs, whether open or not. If we look at the utilization rate of only opened ORs, then for the

Table 4: Comparison of the Deterministic and Stochastic Models

instance	Cancellation Rate		Utilization Rate	
(p-h-d-r)	Deterministic	Stochastic	Deterministic	Stochastic
10-3-5-3	18.1%±1.9E-3	**0.6%**±4.9E-4	5.6%±8.8E-5	**7.7%**±1.1E-4
25-3-5-3	18.9%±1.1E-3	**14.3%**±1.0E-3	15.3%±1.4E-4	**16.2%**±1.5E-4
50-3-5-3	16.4%±7.1E-4	**9.6%**±6.2E-4	31.0%±2.1E-4	**34.5%**±2.3E-4
75-3-5-3	17.8%±5.9E-4	**9.4%**±4.6E-4	45.9%±2.5E-4	**52.0%**±2.8E-4
10-3-5-5	18.2%±1.9E-3	**0.6%**±4.9E-4	3.3%±5.3E-5	**4.6%**±6.8E-5
25-3-5-5	20.9%±1.0E-3	**2.5%**±5.1E-4	8.9%±8.3E-5	**11.3%**±1.1E-4
50-3-5-5	16.5%±7.2E-4	**0.5%**±1.9E-4	18.5%±1.3E-4	**23.1%**±1.6E-4
75-3-5-5	15.3%±5.7E-4	**4.3%**±3.5E-4	28.9%±1.6E-4	**33.2%**±1.8E-4

deterministic model the opened OR utilization rates are in the range of 78% to 83% for the instances in Table 4, while for the stochastic model the opened OR utilization rates are between 67% and 79%. We observe that compared with its deterministic counterpart, the stochastic model has a lower opened OR utilization rate in all the tested instances, because it opens more ORs in order to reduce the cancellation rate.

Notice that all the deterministic models are solved to optimality, while most of the stochastic models have an optimality gap of around 16% when the time limit is reached. Nevertheless, the result from a stochastic model provides a more robust schedule. Therefore, it is worth the effort to implement the stochastic model, even if its implementation is more involved.

7.5 Sensitivity Analysis

In this section we perform several sensitivity analysis for the SDORS problem. Specifically, we change the standard deviation of the surgery duration distribution, the cancellation cost, and the operating time limit of ORs, to see how changes in those parameters influence the performance of our model.

In this experiment it is important to solve all the optimization problems to optimality, because if not, it will be hard to conclude whether the differences in performance are resulted from the change of parameters or differences in optimality gaps. Therefore, all experiments in this section are conducted on instances with $|\mathcal{P}|= 12, |\mathcal{H}|= 2, |\mathcal{D}|= 2, |\mathcal{R}_h|= 2$. Those instances can be solved to optimality within a reasonable time (using the three-stage decomposition), but are also non-trivial enough to produce interesting results, where the change of parameters leads to a change in the measures, such as the cancellation rate and the utilization rate. We use ten instances for each of the different parameter settings.

We first solve the SDORS instances with the original parameter setting ("Baseline"). Then we solve the problem again after one of the following changes:

Case 1: Increase the standard deviation of surgery durations from 40 minutes to 60 minutes.

Case 2: Reduce the unit penalty of canceling patients, κ_3 and κ_4, to $\frac{2}{3}$ of their original values.

Case 3: Reduce the operating time limits of ORs to half of their original values.

After solving those optimization problems and obtain the optimal schedules, we follow the same procedure as in Section 7.4 and obtain the cancellation rate (CancelRate) and utilization rate (UtiliRate) corresponding to each instance. We report the averages and standard deviations of those measure. We also obtain the average numbers and standard deviations of scheduled patients (Scheduled) and opened ORs (OpenOR). Results are shown in Table 5, where the averages of each measure are reported, followed by the standard deviation (STD) in parentheses. We compare the baseline case where none of the parameters are changed, with the three cases that correspond to the three types of changes in the parameters.

Table 5: Sensitivity Analysis for SDORS

Settings	CancelRate	(STD)	UtiliRate	(STD)	Scheduled	(STD)	OpenOR	(STD)
Baseline	8.1%	(2.9E-4)	12.6%	(4.7E-5)	12	(0)	5	(0)
Case 1	8.7%	(1.9E-2)	12.3%	(3.1E-3)	12	(0)	5	(3.2E-1)
Case 2	16.7%	(4.7E-4)	11.3%	(5.6E-5)	12	(0)	4	(0)
Case 3	5.1%	(2.9E-4)	17.1%	(7.5E-5)	8	(0)	8	(0)

From the results, we see that for Case 1 where the standard deviation of the surgery duration becomes higher, we have a higher cancellation rate and a lower OR utilization rate. This is expected, as more uncertainty usually leads to more cancellations. Also, in our experiment, the average number of scheduled patients and the average number of opened ORs are the same for Baseline and Case 1, so a higher cancellation rate naturally leads to a lower OR utilization.

For Case 2, the cancellation rate increases significantly, and the OR utilization rate drops. When cancelling a patient becomes less costly, the hospitals are motivated to schedule patients to fewer ORs to save operational costs. In the experiment, the average number of opened ORs drops from 5 (Baseline) to 4, while the average number of scheduled patients remains the same. This trend explains the increase in the cancellation rate. Also, compared with the drop in opened ORs, the increase in cancellation rate is more drastic, and this leads to an overall effect of a decreased utilization rate.

For Case 3, the cancellation rate decreases and the OR utilization rate increases. When the opening hours of ORs become shorter, the average number of patients per OR becomes lower. This understandably reduces cancellation, as fewer patients per OR means lower uncertainty. The decrease in the cancellation rate may explain the increase in the OR utilization rate.

8 Conclusions

In this paper we propose the SDORS problem. Through computational experiments we show that, when compared with its deterministic counterpart from the literature, the SDORS model generates more robust schedules that reduce the rate of cancellation and improve the rate of OR utilization.

We also develop several decomposition algorithms and algorithmic enhancements to improve the computational efficiency, and conduct numerical experiments on real instances from the UHN.

For all the instances we experimented with, our algorithms are able to reduce the solution time or the optimality gap compared with the commercial solver. Moreover, our algorithms have the potential to be applied to any stochastic distributed bin packing problem, as the SDORS problem can be viewed as an extension of a stochastic distributed bin packing problem.

We recognize that the SDORS problem is a very difficult problem to solve. Although our algorithms perform much better than the commercial solver, there is still room for improvement, to be able to efficiently handle practical instances, where there are hundreds of patients to be scheduled each week. Therefore, future works that further improve the computational efficiency will be valuable.

References

[1] G. Angulo, S. Ahmed, and S. S. Dey. Improving the integer L-shaped method. *INFORMS J. Comput.*, 28(3):483–499, 2016.

[2] E. Balas, S. Ceria, and G. Cornuéjols. A lift-and-project cutting plane algorithm for mixed 0–1 programs. *Math. Program.*, 58(1-3):295–324, 1993.

[3] B. Barua and D. Jacques. The private cost of public queues for medically necessary care. Technical report, Fraser Institute, 2019.

[4] J. Behnamian and S. F. Ghomi. The heterogeneous multi-factory production network scheduling with adaptive communication policy and parallel machine. *Inf. Sci.*, 219:181–196, 2013.

[5] E. Beier, S. Venkatachalam, L. Corolli, and L. Ntaimo. Stage-and scenario-wise fenchel decomposition for stochastic mixed 0-1 programs with special structure. *Comput. and Oper. Res.*, 59:94–103, 2015.

[6] J. Beliën and E. Demeulemeester. A branch-and-price approach for integrating nurse and surgery scheduling. *European J. Oper. Res.*, 189(3):652–668, 2008.

[7] J. F. Benders. Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik*, 4(1):238–252, 1962.

[8] D. Bergman, A. A. Cire, W.-J. Van Hoeve, and J. Hooker. *Decision diagrams for optimization*. Springer, 2016.

[9] J. T. Blake and M. W. Carter. A goal programming approach to strategic resource allocation in acute care hospitals. *European J. Oper. Res.*, 140(3):541–561, 2002.

[10] J. T. Blake and J. Donald. Mount Sinai hospital uses integer programming to allocate operating room time. *Interfaces*, 32(2):63–73, 2002.

[11] M. Bodur. *On valid inequalities for polyhedra in extended and projected spaces with application to two-stage stochastic integer programming*. PhD thesis, The University of Wisconsin-Madison, 2015.

[12] M. Bodur and J. R. Luedtke. Mixed-integer rounding enhanced Benders decomposition for multiclass service-system staffing and scheduling with arrival rate uncertainty. *Management Sci.*, 63(7):2073–2091, 2016.

[13] J. Bowers and G. Mould. Managing uncertainty in orthopaedic trauma theatres. *European J. Oper. Res.*, 154(3):599–608, 2004.

[14] B. Cardoen, E. Demeulemeester, and J. Beliën. Operating room planning and scheduling: A literature review. *European J. Oper. Res.*, 201(3):921–932, 2010.

[15] C. C. Carøe and J. Tind. L-shaped decomposition of two-stage stochastic programs with integer recourse. *Math. Program.*, 83(1-3):451–464, 1998.

[16] CIHI. Wait times for priority procedures in Canada. https://www.cihi.ca/en/wait-times-for-priority-procedures-in-canada, 2020. [Online; accessed October 2nd, 2020].

[17] A. A. Ciré, E. Coban, and J. N. Hooker. Logic-based Benders decomposition for planning and scheduling: a computational analysis. *The Knowledge Engineering Review*, 31(5):440–451, 2016.

[18] Y. Deng, S. Shen, and B. Denton. Chance-constrained surgery planning under conditions of limited and ambiguous data. *INFORMS J. Comput.*, 31(3):559–575, 2019.

[19] B. T. Denton, A. J. Miller, H. J. Balasubramanian, and T. R. Huschka. Optimal allocation of surgery blocks to operating rooms under uncertainty. *Oper. Res.*, 58(4-part-1):802–816, 2010.

[20] P. Dimitriadis, S. Iyer, and E. Evgeniou. The challenge of cancellations on the day of surgery. *International J. Surg.*, 11(10):1126–1130, 2013.

[21] M. M. Fazel-Zarandi, O. Berman, and J. C. Beck. Solving a stochastic facility location/fleet management problem with logic-based Benders' decomposition. *IIE Trans.*, 45(8):896–911, 2013.

[22] H. Fei, C. Chu, N. Meskens, and A. Artiba. Solving surgical cases assignment problem by a branch-and-price approach. *International J. Production Economics*, 112(1):96–108, 2008.

[23] H. Fei, C. Chu, and N. Meskens. Solving a tactical operating room planning problem by a column-generation-based heuristic procedure with four criteria. *European J. Oper. Res.*, 166 (1):91, 2009.

[24] D. Gade, S. Küçükyavuz, and S. Sen. Decomposition algorithms with parametric gomory cuts for two-stage stochastic integer programs. *Math. Program.*, 144(1-2):39–64, 2014.

[25] F. Guerriero and R. Guido. Operational research in the management of the operating theatre: a survey. *Health Care Manage Sci.*, 14(1):89–114, 2011.

[26] S. Gul, B. T. Denton, and J. W. Fowler. A progressive hedging approach for surgery planning under uncertainty. *INFORMS J. Comput.*, 27(4):755–772, 2015.

[27] P. R. Harper. A framework for operational modelling of hospital resources. *Health Care Manage Sci.*, 5(3):165–173, 2002.

[28] J. N. Hooker and G. Ottosson. Logic-based Benders decomposition. *Math. Program.*, 96(1): 33–60, 2003.

[29] A. Jebali, A. B. H. Alouane, and P. Ladet. Operating rooms scheduling. *International J. Production Economics*, 99(1-2):52–62, 2006.

[30] D. S. Johnson, A. Demers, J. D. Ullman, M. R. Garey, and R. L. Graham. Worst-case performance bounds for simple one-dimensional packing algorithms. *SIAM J. Comput.*, 3(4): 299–325, 1974.

[31] M. H. Karwan. *Surrogate constraint duality and extensions in integer programming.* PhD thesis, Georgia Institute of Technology, 1976.

[32] G. Laporte and F. V. Louveaux. The integer L-shaped method for stochastic integer programs with complete recourse. *Oper. Res. Lett.*, 13(3):133–142, 1993.

[33] M. Lombardi, M. Milano, M. Ruggiero, and L. Benini. Stochastic allocation and scheduling for conditional task graphs in multi-processor systems-on-chip. *J. Sched.*, 13(4):315–345, 2010.

[34] L. Lozano and J. C. Smith. A binary decision diagram based algorithm for solving a class of binary two-stage stochastic programs. *Math. Program.*, pages 1–24, 2018.

[35] J. Magnussen, T. P. Hagen, and O. M. Kaarboe. Centralized or decentralized? a case study of norwegian hospital reform. *Social science & medicine*, 64(10):2129–2137, 2007.

[36] I. Marques, M. E. Captivo, and M. V. Pato. An integer programming approach to elective surgery scheduling. *OR Spectrum*, 34(2):407–427, 2012.

[37] D. Min and Y. Yih. Scheduling elective surgery under uncertainty and downstream capacity constraints. *European J. Oper. Res.*, 206(3):642–652, 2010.

[38] B. Naderi and R. Ruiz. A scatter search algorithm for the distributed permutation flowshop scheduling problem. *European J. Oper. Res.*, 239(2):323–334, 2014.

[39] L. Ntaimo. Fenchel decomposition for stochastic mixed-integer programming. *J. Global. Opt.*, 55(1):141–163, 2013.

[40] L. Ntaimo and M. W. Tanner. Computations with disjunctive cuts for two-stage stochastic mixed 0-1 integer programs. *J. Global. Opt.*, 41(3):365–384, 2008.

[41] V. Roshanaei, C. Luong, D. M. Aleman, and D. Urbach. Propagating logic-based Benders decomposition approaches for distributed operating room scheduling. *European J. Oper. Res.*, 257(2):439–455, 2017.

[42] V. Roshanaei, C. Luong, D. M. Aleman, and D. R. Urbach. Collaborative operating room planning and scheduling. *INFORMS J. Comput.*, 29(3):558–580, 2017.

[43] H. D. Sherali and B. M. Fraticelli. A modification of benders' decomposition algorithm for discrete subproblems: An approach for stochastic programs with integer recourse. *J. Global. Opt.*, 22(1-4):319–342, 2002.

[44] D. P. Strum, J. H. May, and L. G. Vargas. Modeling the uncertainty of surgical procedure timescomparison of log-normal and normal models. *Anesthesiol.: J. Am. Soc. Anesthesiol.*, 92 (4):1160–1167, 2000.

[45] C. H. Timpe and J. Kallrath. Optimal planning in large multi-site production networks. *European J. Oper. Res.*, 126(2):422–435, 2000.

[46] S. Wang, V. Roshanaei, D. Aleman, and D. Urbach. A discrete event simulation evaluation of distributed operating room scheduling. *IIE Trans. Healthc. Syst. Engineering*, 6(4):236–245, 2016.

A Proofs

A.1 Proof of Theorem 1

Theorem 5. *The LBBD optimality cut (6), which is defined as*

$$Q^s_{hdr} \geq \bar{Q}^s_{hdr} - \sum_{p \in \hat{\mathcal{P}}_{hdr}} c_p^{cancel}(1 - x_{hdpr}),$$

is valid.

Proof. Proof. Note that the cut (6) is formulated for one OR, r, in one hospital h on one day d for scenario s, thus the following argument is formed in terms of a single tuple of (h, d, r, s) and holds for each such tuple.

To prove the validity of the cut, we need to ensure two points: the cut eliminates the current master solution, and it does not exclude any globally optimal solution.

Let us first prove that the current master solution ($\hat{u}_{hd}$, $\hat{y}_{hdr}$, $\hat{x}_{hdpr}$, $\hat{w}_p$, $\hat{Q}^s_{hdr}$) violates the cut, i.e., it will be cut off. Assume towards contradiction that it satisfies the cut. When we substitute the master solution to the cut (6), the left-hand side (LHS) of the cut becomes $\hat{Q}^s_{hdr}$, while the right-hand side (RHS) of the cut becomes $\bar{Q}^s_{hdr}$ as all $\hat{x}_{hdpr}$'s in the set $\hat{P}_{hdr}$ have the value 1. This gives us $\hat{Q}^s_{hdr} \geq \bar{Q}^s_{hdr}$. However, we know that $\hat{Q}^s_{hdr} < \bar{Q}^s_{hdr}$ because otherwise the LBBD optimality cut will not be generated. This is a contradiction, therefore the current master solution does not satisfy the cut.

Next, we need to prove that any globally optimal solution, denoted by ($\hat{u}^*_{hd}$, $\hat{y}^*_{hdr}$, $\hat{x}^*_{hdpr}$, $\hat{w}^*_p$, $\bar{Q}^{s*}_{hdr}$), is not excluded by the LBBD optimality cut that is generated from a master solution ($\hat{u}_{hd}$, $\hat{y}_{hdr}$, $\hat{x}_{hdpr}$, $\hat{w}_p$, $\hat{Q}^s_{hdr}$) using $\hat{P}_{hdr}$. Here, $(\hat{u}^*_{hd}, \hat{y}^*_{hdr}, \hat{x}^*_{hdpr}, \hat{w}^*_p)$ is obtained by solving the master problem, and $\bar{Q}^{s*}_{hdr}$ is the corresponding optimal objective value of the subproblem. For a globally optimal solution, the cancellation cost obtained by the master problem, $\hat{Q}^{s*}_{hdr}$, should match that from the subproblem, $\bar{Q}^{s*}_{hdr}$.

We discuss the different cases of globally optimal solutions below:

Case 1: If $\hat{y}^*_{hdr} = 0$, i.e. the OR r in hospital h on day d is closed, then the optimal solution must have $\hat{x}^*_{hdpr} = 0, \forall p \in \mathcal{P}$ due to (1g). This in turn yields all $z^*_p = 0$ in the optimal subproblem solution due to (5c) and thus $\bar{Q}^{s*}_{hdr} = 0$. Replacing those values in the cut, we get the following:

$$\overbrace{\bar{Q}^{s*}_{hdr}}^{=0} \geq \bar{Q}^s_{hdr} - \overbrace{\sum_{p \in \hat{\mathcal{P}}_{hdr}} c^{\text{cancel}}_p (1 - x^*_{hdpr})}^{=\sum_{p \in \hat{\mathcal{P}}_{hdr}} c^{\text{cancel}}_p} \qquad (\star)$$

As $\bar{Q}^s_{hdr}$ is the cancellation cost for the (h, d, r, s) tuple corresponding to the patient list $\hat{\mathcal{P}}_{hdr}$, we have

$$\bar{Q}^s_{hdr} \leq \sum_{p \in \hat{\mathcal{P}}_{hdr}} c^{\text{cancel}}_p$$

which follows from the fact that the cancellation cost of an OR cannot exceed the total cancellation cost of all patients assigned to it. We can now conclude that the RHS of ($\star$) is nonpositive. Therefore, the global optimal solution $(\hat{u}^*_{hd}, \hat{y}^*_{hdr}, \hat{x}^*_{hdpr}, \hat{w}^*_p, \bar{Q}^{s*}_{hdr})$ satisfies the LBBD optimality cut.

Case 2: If $\hat{y}^*_{hdr} = 1$, then we must have some patients assigned to the (h, d, r) tuple, otherwise we can close this OR and save the cost. At the optimal patient assignment, $\hat{x}^*_{hdpr}$, we either still have all the patients in the set $\hat{\mathcal{P}}_{hdr}$ in the (h, d, r) tuple, or some patients are no longer assigned to this (h, d, r) tuple. We further discuss those two cases separately:

Subcase a: If all the patients in $\hat{\mathcal{P}}_{hdr}$ are assigned to the current (h, d, r) tuple at an optimal solution, i.e., $\hat{x}^*_{hdpr} = 1, \forall p \in \hat{\mathcal{P}}_{hdr}$, then the corresponding cancellation cost for the current patient list, $\bar{Q}^{s*}_{hdr}$, should not be lower than the cancellation cost for $\hat{\mathcal{P}}_{hdr}$, which is $\bar{Q}^s_{hdr}$. That is, we have $\bar{Q}^{s*}_{hdr} \geq \bar{Q}^s_{hdr}$. Substitute the optimal solution into the LBBD cut:

$$\overbrace{\bar{Q}^{s*}_{hdr}}^{\geq \bar{Q}^s_{hdr}} \geq \bar{Q}^s_{hdr} - \sum_{p \in \hat{\mathcal{P}}_{hdr}} c^{\text{cancel}}_p \overbrace{(1 - \hat{x}^*_{hdpr})}^{=0}$$

which holds.

Subcase b: If at the global optimal solution only a subset of patients in $\hat{\mathcal{P}}_{hdr}$ are still assigned to the current (h, d, r), the proof is more involved. For the ease of proof, we introduce some more notations. Let us denote the patients from $\hat{\mathcal{P}}_{hdr}$ who are still assigned by set $\hat{\mathcal{P}}^{A*}_{hdr} \subset \hat{\mathcal{P}}_{hdr}$, and patients in $\hat{\mathcal{P}}_{hdr}$ who are no longer assigned as $\hat{\mathcal{P}}^{N*}_{hdr} = \hat{\mathcal{P}}_{hdr} \setminus \hat{\mathcal{P}}^{A*}_{hdr}$. Also, in the global optimal solution there may exist patients who are assigned to the current (h, d, r) but do not belong to $\hat{\mathcal{P}}_{hdr}$, we denote those patients by $\tilde{\mathcal{P}}^{A*}_{hdr} \subseteq \mathcal{P}_{hdr} \setminus \hat{\mathcal{P}}_{hdr}$. Then the set of assigned patients in the global optimal solution is $\mathcal{P}^{A*}_{hdr} = \hat{\mathcal{P}}^{A*}_{hdr} \cup \tilde{\mathcal{P}}^{A*}_{hdr}$. As noted before, the optimal cancellation cost corresponding to the assignment of $\mathcal{P}^{A*}_{hdr}$ is $\bar{Q}^{s*}_{hdr}$, while the cancellation cost corresponding to $\hat{\mathcal{P}}_{hdr}$ is $\bar{Q}^{s}_{hdr}$. It is also useful to find the cancellation cost when only patients in $\hat{\mathcal{P}}^{A*}_{hdr}$ are assigned to the (h, d, r, s) tuple, which we denote by $\bar{Q}^{s}_{hdr}(\hat{\mathcal{P}}^{A*}_{hdr})$. It is easy to see that $\bar{Q}^{s*}_{hdr} \geq \bar{Q}^{s}_{hdr}(\hat{\mathcal{P}}^{A*}_{hdr})$, as $\hat{\mathcal{P}}^{A*}_{hdr}$ is a subset of $\mathcal{P}^{A*}_{hdr}$.

To help illustrate the relationships between patient sets, we use the following simple example in Figure 2. There is a set of six patients $\{p_1, ..., p_6\} \in \mathcal{P}$. The first three of those patients are scheduled by the current master solution to the current (h, d, r), i.e. $\{p_1, p_2, p_3\} \in \hat{\mathcal{P}}_{hdr}$. In the figure they are colored with gray. The global optimal solution schedules the patient set $\mathcal{P}^{A*}_{hdr} = \{p_2, p_3, p_4, p_5\}$. Then according to our definition, $\hat{\mathcal{P}}^{A*}_{hdr} = \mathcal{P}^{A*}_{hdr} \cap \hat{\mathcal{P}}_{hdr} = \{p_2, p_3\}$, $\tilde{\mathcal{P}}^{A*}_{hdr} = \mathcal{P}^{A*}_{hdr} \setminus \hat{\mathcal{P}}^{A*}_{hdr} = \{p_4, p_5\}$, and $\hat{\mathcal{P}}^{N*}_{hdr} = \hat{\mathcal{P}}_{hdr} \setminus \hat{\mathcal{P}}^{A*}_{hdr} = \{p_1\}$.

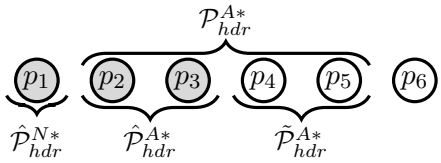

Figure 2: Illustration of relationships between patient sets

We claim that $\bar{Q}^{s}_{hdr}(\hat{\mathcal{P}}^{A*}_{hdr}) \geq \bar{Q}^{s}_{hdr} - \sum_{p \in \hat{\mathcal{P}}^{N*}_{hdr}} c^{\text{cancel}}_{p}$. To prove this, assume towards contradiction that it is not true. Then we have $\bar{Q}^{s}_{hdr}(\hat{\mathcal{P}}^{A*}_{hdr}) < \bar{Q}^{s}_{hdr} - \sum_{p \in \hat{\mathcal{P}}^{N*}_{hdr}} c^{\text{cancel}}_{p}$, meaning that if in the current OR we have all patients from $\mathcal{P}^{A*}_{hdr}$, then some patients in $\mathcal{P}^{N*}_{hdr}$ are also scheduled to this OR, the cancellation cost can increase for at most $\sum_{p \in \hat{\mathcal{P}}^{N*}_{hdr}} c^{\text{cancel}}_{p}$. If that is true, then the cancellation cost for the assignment $\hat{\mathcal{P}}^{A*}_{hdr} \cup \hat{\mathcal{P}}^{N*}_{hdr}$ is at most $\bar{Q}^{s}_{hdr}(\hat{\mathcal{P}}^{A*}_{hdr}) + \sum_{p \in \hat{\mathcal{P}}^{N*}_{hdr}} c^{\text{cancel}}_{p} < \bar{Q}^{s}_{hdr}$. However, the patient set $\hat{\mathcal{P}}^{A*}_{hdr} \cup \hat{\mathcal{P}}^{N*}_{hdr}$ is equivalent to the assignment with patients in the set $\hat{P}_{hdr}$, and its corresponding cancellation cost is exactly $\bar{Q}^{s}_{hdr}$. This is a contradiction.

As the lower bound of $\bar{Q}^{s}_{hdr}(\hat{\mathcal{P}}^{A*}_{hdr})$ is $\bar{Q}^{s}_{hdr} - \sum_{p \in \hat{\mathcal{P}}^{N*}_{hdr}} c^{\text{cancel}}_{p}$, we have the following evaluation of the LBBD cut at the global optimal optimization:

$$\bar{Q}^{s*}_{hdr} \geq \bar{Q}^{s}_{hdr}(\hat{\mathcal{P}}^{A*}_{hdr}) \geq \bar{Q}^{s}_{hdr} - \sum_{p \in \hat{\mathcal{P}}^{N}_{hdr}} c^{\text{cancel}}_{p} \overbrace{(1 - \hat{x}^{*}_{hdp})}^{=1} - \sum_{p \in \hat{\mathcal{P}}^{A}_{hdr}} c^{\text{cancel}}_{p} \overbrace{(1 - \hat{x}^{*}_{hdp})}^{=0}$$

which is satisfied thanks to the relations of the cut's LHS and RHS to the middle comparative term as mentioned above. □ □

A.2 Proof of Theorem 2

Theorem 6. *Constraints* (13)*, which are defined as*

$$Q^s_{hdr} \geq \left(\min_{p\in\mathcal{P}} \frac{c_p^{cancel}}{T_p^s}\right)\left(\sum_{p\in\mathcal{P}} T_p^s x_{hdpr} - B_{hd}\right) \quad \forall h \in \mathcal{H}, d \in \mathcal{D}, r \in \mathcal{R}_h, s \in \mathcal{S},$$

provide valid lower bounds (LBs) for Q^s_{hdr}.

Proof. Proof. At optimality Q^s_{hdr} equals the optimal objective value of the subproblem (5). We want to prove that the RHS of inequality (13) is either trivially true or otherwise can be obtained by relaxing the subproblem.

First, we look at the trivial case where the OR corresponding to Q^s_{hdr} is not opened. In this case, there should be no cost for cancelling as no patient is assigned in the first place. Therefore, $Q^s_{hdr} = 0$. Since in this case the RHS of the inequality becomes $\min_{p\in\mathcal{P}}\left(\frac{c_p^{\text{cancel}}}{T_p^s}\right)(-B_{hd}) < 0$ as the consequence of $x_{hdpr} = 0$ $(\forall p \in \mathcal{P})$, the constraint is valid.

If the OR corresponding to Q^s_{hdr} is opened in an optimal solution, there must exist at least one patient who is assigned to this OR. Given an assignment of patients, $\hat{x}_{hdpr}$ $(\forall p \in \mathcal{P})$, and the set of assigned patients, $\hat{\mathcal{P}}_{hdr}$, suppose a subset of the patients, $\hat{\mathcal{P}}^C_{hdr} \subseteq \hat{\mathcal{P}}_{hdr}$, is cancelled as dictated by the optimal solution of the subproblem (5). Then by definition we have $Q^s_{hdr} = \sum_{p\in\hat{\mathcal{P}}^C_{hdr}} c_p^{\text{cancel}}$. Due to the fact that after cancellation, the total surgery duration of accepted patients should be no more than the operating time limit, B_{hd}, we have the following:

$$\begin{aligned}
\sum_{p\in\hat{\mathcal{P}}^C_{hdr}} c_p^{\text{cancel}} &= \sum_{p\in\hat{\mathcal{P}}^C_{hdr}} c_p^{\text{cancel}} \hat{x}_{hdpr} \\
&= \sum_{p\in\hat{\mathcal{P}}^C_{hdr}} \frac{c_p^{\text{cancel}}}{T_p^s} T_p^s \hat{x}_{hdpr} \\
&\geq \left(\min_{p\in\mathcal{P}} \frac{c_p^{\text{cancel}}}{T_p^s}\right) \sum_{p\in\hat{\mathcal{P}}^C_{hdr}} T_p^s \hat{x}_{hdpr} \\
&\geq \left(\min_{p\in\mathcal{P}} \frac{c_p^{\text{cancel}}}{T_p^s}\right)\left(\sum_{p\in\hat{\mathcal{P}}^C_{hdr}} T_p^s \hat{x}_{hdpr} + \left(\sum_{p\in\mathcal{P}\setminus\hat{\mathcal{P}}^C_{hdr}} T_p^s \hat{x}_{hdpr} - B_{hd}\right)\right) \\
&= \left(\min_{p\in\mathcal{P}} \frac{c_p^{\text{cancel}}}{T_p^s}\right)\left(\sum_{p\in\mathcal{P}} T_p^s \hat{x}_{hdpr} - B_{hd}\right)
\end{aligned}$$

Therefore, for any assignment $\hat{x}_{hdpr}$ $(\forall p \in \mathcal{P})$, the expression $\left(\min_{p\in\mathcal{P}} \frac{c_p^{\text{cancel}}}{T_p^s}\right)\left(\sum_{p\in\mathcal{P}} T_p^s \hat{x}_{hdpr} - B_{hd}\right)$ provides a LB for Q^s_{hdr}. Substitute the fixed assignment $\hat{x}_{hdpr}$ with the variable x_{hdpr} and we get the constraint (13). □ □

A.3 Proof of Theorem 3

Theorem 7. *The LBBD optimality cut (16), which is defined as*

$$
\begin{aligned}
&Q_{hd} \geq \bar{Q}_{hd} \left(g_{hdj} - \sum_{p \in \hat{\mathcal{P}}_{hd}} (1 - x_{hdp}) \right) \\
&y_{hd} \geq (1 + \hat{y}_{hd})(1 - g_{hdj}), \\
&g_{hdj} \in \{0, 1\}
\end{aligned}
$$

is valid, where $\bar{Q}_{hd}$ is the optimal objective value of the LBBD subproblem (15), and $\hat{y}_{hd}$ is the optimal solution of y_{hd} from the LBBD master problem (14).

Proof. Proof. For any (h, d) pair, we need to prove that any global optimal solution $(\hat{u}'_{hd}, \hat{y}'_{hd}, \hat{x}'_{hdp}, \bar{Q}'_{hd})$ is not excluded by the optimality cut. Given an optimal (DE) solution, $(\hat{u}'_{hd}, \hat{y}'_{hd}, \hat{x}'_{hdp})$ denotes the corresponding solutions for the main LBBD master decisions, and $\bar{Q}'_{hd}$ is the corresponding optimal objective of the subproblem.

Case 1: If $\hat{u}'_{hd} = 0$, then the only global optimal solution for the current h and d will be $\hat{y}'_{hd} = 0, \hat{x}'_{hdp} = 0, \bar{Q}'_{hd} = 0$. This is feasible to the optimality cut.

Case 2: If $\hat{u}'_{hd} = 1$, then we discuss the following two cases separately:

Subcase a: If $\hat{y}'_{hd} > \hat{y}_{hd}$, we are not able to find a nontrivial LB for Q_{hd} without solving another subproblem, because when there are more ORs available, the cancellation cost of the current (h, d) pair can be either zero or some nonzero value that is smaller than $\bar{Q}_{hd}$. This is why we make the cut redundant in this case: let $g_{hdj} = 0$, (16a) becomes

$$
Q_{hd} \geq \bar{Q}_{hd} \left(- \sum_{p \in \hat{P}_{hd}} (1 - x_{hdp}) \right)
$$

This is always true since RHS is nonpositive, so $\bar{Q}'_{hd}$ and $\hat{x}'_{hdp}$ also satisfy this inequality. (16b) is now $y_{hd} \geq (1 + \hat{y}_{hd})$, which is equivalent to $\hat{y}_{hd} > \hat{y}_{hd}$, and that is exactly the assumption of this case.

Subcase b: If $\hat{y}'_{hd} \leq \hat{y}_{hd}$, then there are further two subcases to discuss. Note that due to (16b), we always have $g_{hdj} = 1$ in this case.

Subcase b1: If in the global optimal all patients that are assigned in the current solution are still assigned, i.e. $\hat{x}'_{hdp} \geq \hat{x}_{hdp}, \forall p \in \mathcal{P}_{hd}$, then we claim that the optimal cancellation cost will not decrease from the current value $\bar{Q}_{hd}$ because there are the same number of or a smaller number of rooms, but all the current assigned patients are still assigned. To see this argument is another way, assume for contradiction that $Q_{hd} < \bar{Q}_{hd}$ in this case. Then this means we can also schedule the patients in the current patient list in the same way with a lower cancellation cost than $\bar{Q}_{hd}$, which is a contradiction. Therefore, $Q_{hd} \geq \bar{Q}_{hd}$. Let $g_{hdj} = 1$, and also replace Q_{hd} and x_{hdp} with $\hat{Q}'_{hdp}$

and $\hat{x}'_{hdp}$, (16) becomes:

$$\overbrace{\hat{Q}'_{hd}}^{\geq \bar{Q}_{hd}} \geq \bar{Q}_{hd}\left(\overbrace{g_{hdj}}^{=1} - \overbrace{\sum_{p\in\hat{P}_{hd}}(1-\hat{x}'_{hdp})}^{=0}\right)$$
$$\overbrace{\hat{y}'_{hd}}^{\leq \hat{y}_{hd}} \geq (1+\hat{y}_{hd})(1-\overbrace{g_{hdj}}^{=1})$$

Therefore, in this case the global solution also satisfies the optimality cut.

Subcase b2: If some currently assigned patients are no longer assigned, then we cannot give a nontrivial LB for the global optimal $\bar{Q}'_{hd}$, because the cancellation cost can either be zero or be a nonzero value that is larger or smaller than $\bar{Q}_{hd}$. Thus we make the LBBD cut redundant:

$$\overbrace{\hat{Q}'_{hd}}^{\geq 0} \geq \bar{Q}_{hd}\left(\overbrace{g_{hdj}}^{=1} - \overbrace{\sum_{p\in\hat{P}_{hd}}(1-\hat{x}'_{hdp})}^{\geq 1}\right)$$
$$\overbrace{\hat{y}'_{hd}}^{\leq \hat{y}_{hd}} \geq (1+\hat{y}_{hd})(1-\overbrace{g_{hdj}}^{=1})$$

which is satisfied by the global optimal solution. □ □

A.4 Proof of Theorem 4

Theorem 8. *The constraints* (20)*, which are defined as*

$$Q_{hd} \geq \frac{1}{|\mathcal{S}|}\sum_{s\in\mathcal{S}}\left(\min_{p\in\mathcal{P}}\frac{c_p^{cancel}}{T_p^s}\right)\left(\sum_{p\in\mathcal{P}}T_p^s x_{hdp} - B_{hd}y_{hd}\right) \quad \forall h\in\mathcal{H}, d\in\mathcal{D},$$

are valid for (DE).

Proof. Proof. At optimality Q_{hd} equals the optimal objective value of the subproblem (15). We want to prove that the RHS of inequality (20) is either trivially true or otherwise can be obtained by relaxing the subproblem.

First, we look at the trivia case where the (h,d) pair corresponding to Q_{hd} is not opened. In this case, there should be no cost for cancelling as no patient is assigned in the first place. Therefore, $Q_{hd}=0$. Since in this case the RHS of the inequality becomes 0 as the consequence of $x_{hdp}=0$ $(\forall p\in\mathcal{P})$ and $y_{hd}=0$, the constraint is valid.

If the (h,d) pair corresponding to Q_{hd} is opened in an optimal solution, there must exist at least one patient who is assigned to this (h,d) pair. Given an assignment of patients, $\hat{x}_{hdp}$ $(\forall p\in\mathcal{P})$, the set of assigned patients, $\hat{\mathcal{P}}_{hd}$, and the number of opened ORs, $\hat{y}_{hd}$. Suppose under the scenario s a set of patients, $\hat{\mathcal{P}}^{sC}_{hdr}\subseteq\hat{\mathcal{P}}_{hd}$, is cancelled as dictated by the optimal solution of the subproblem (15). Then by definition we have $Q_{hd}=\frac{1}{|\mathcal{S}|}\sum_{s\in\mathcal{S}}\sum_{p\in\hat{\mathcal{P}}^{sC}_{hdr}}c_p^{\text{cancel}}$. Due to the fact that after cancellation,

the total surgery duration of accepted patients should be no more than the total operating time limit of opened ORs, $B_{hd}\hat{y}_{hd}$, we have the following for any scenario $s \in \mathcal{S}$:

$$\begin{aligned}
\sum_{p\in\hat{\mathcal{P}}^{sC}_{hdr}} c_p^{\text{cancel}} &= \sum_{p\in\hat{\mathcal{P}}^{sC}_{hdr}} c_p^{\text{cancel}}\hat{x}_{hdp} \\
&= \sum_{p\in\hat{\mathcal{P}}^{sC}_{hdr}} \frac{c_p^{\text{cancel}}}{T_p^s} T_p^s\hat{x}_{hdp} \\
&\geq \left(\min_{p\in\mathcal{P}} \frac{c_p^{\text{cancel}}}{T_p^s}\right) \sum_{p\in\hat{\mathcal{P}}^{sC}_{hdr}} T_p^s\hat{x}_{hdp} \\
&\geq \left(\min_{p\in\mathcal{P}} \frac{c_p^{\text{cancel}}}{T_p^s}\right) \left(\sum_{p\in\hat{\mathcal{P}}^{sC}_{hdr}} T_p^s\hat{x}_{hdp} + \left(\sum_{p\in\mathcal{P}\setminus\hat{\mathcal{P}}^{sC}_{hdr}} T_p^s\hat{x}_{hdp} - B_{hd}\hat{y}_{hd}\right)\right) \\
&= \left(\min_{p\in\mathcal{P}} \frac{c_p^{\text{cancel}}}{T_p^s}\right) \left(\sum_{p\in\mathcal{P}} T_p^s\hat{x}_{hdp} - B_{hd}\hat{y}_{hd}\right)
\end{aligned}$$

Therefore, for any assignment $\hat{x}_{hdp}$ ($\forall p \in \mathcal{P}$), the expression $\frac{1}{|\mathcal{S}|}\sum_{s\in\mathcal{S}} \left(\min_{p\in\mathcal{P}} \frac{c_p^{\text{cancel}}}{T_p^s}\right)\left(\sum_{p\in\mathcal{P}} T_p^s\hat{x}_{hdp} - B_{hd}\hat{y}_{hd}\right)$ provides a LB for Q_{hd}. Substitute the fixed assignment $\hat{x}_{hdp}$ with the variable x_{hdp} and $\hat{y}_{hd}$ with y_{hd} then we get the constraint (20). □ □

B Flow Chart for the Two-stage Decomposition

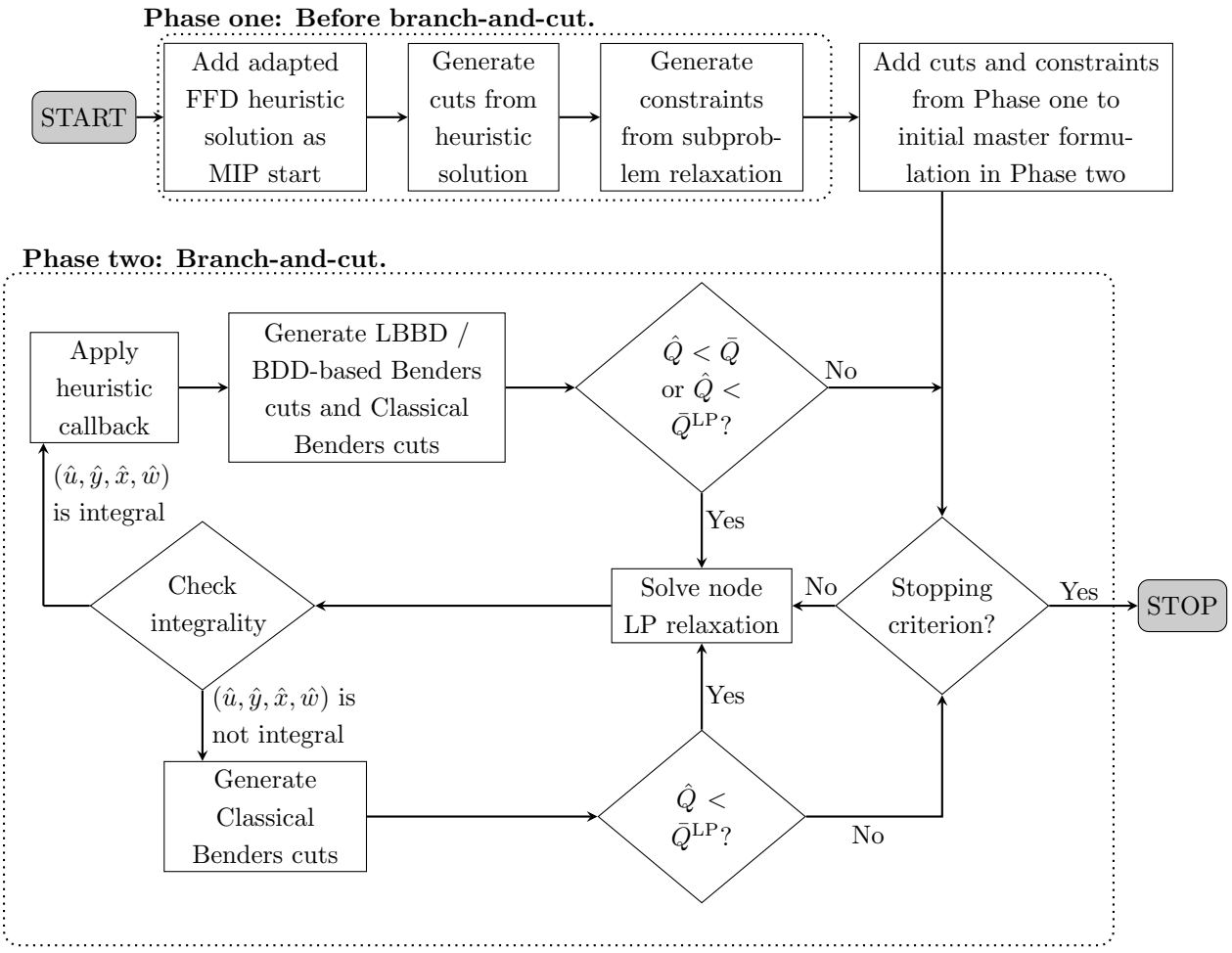

Figure 3: Flow chart of the two-stage decomposition algorithm.

C Classical Benders Cuts for the Three-stage Decomposition

For the LBBD subproblem, we relax the LBBD subproblem (15). The binary variables x_{pr}, y_r, and z^s_{pr} are redefined as continuous variables. The variables in the parentheses at the end of constraints some constraints are their corresponding dual variables. We denote the relaxed LBBD subproblem by $\mathcal{Q}^{\text{LP}}_{hd}(\hat{x}_{hd\cdot}, \hat{y}_{hd}, T^s_{\cdot})$, and shorten it as $\bar{Q}^{\text{LP}}_{hd}$ later in the text:

$$\mathcal{Q}^{\text{LP}}_{hd}(\hat{x}_{hd\cdot}, \hat{y}_{hd}, T^s_{\cdot}) = \min \ \frac{1}{|\mathcal{S}|} \sum_{s\in\mathcal{S}} \sum_{p\in\mathcal{P}} \sum_{r\in\mathcal{R}_h} c_p(x_{pr} - z^s_{pr})$$

$$\text{s.t.} \ \sum_{r\in\mathcal{R}_h} x_{pr} = \hat{x}_{hdp} \qquad \forall p \in \mathcal{P} \qquad (\gamma_p)$$

$$\sum_{p\in\mathcal{P}} T_p^s z_{pr}^s \le B_{hd} y_r \qquad \forall r \in \mathcal{R}_h, s \in \mathcal{S}$$

$$z_{pr}^s \le x_{pr} \qquad \forall p \in \mathcal{P}, r \in \mathcal{R}_h, s \in \mathcal{S}$$

$$x_{pr} \le y_r \qquad \forall p \in \mathcal{P}, r \in \mathcal{R}_h$$

$$\sum_{r\in\mathcal{R}_h} y_r \le \hat{y}_{hd} \qquad (\beta)$$

$$y_r \le 1 \qquad \forall r \in \mathcal{R}_h \qquad (\delta_r)$$

$$y_r, x_{pr}, z_{pr}^s \ge 0 \qquad \forall p \in \mathcal{P}, r \in \mathcal{R}_h, s \in \mathcal{S}$$

When $\hat{Q}_{hd} < \bar{Q}_{hd}^{\mathrm{LP}}$, the following classical Benders cuts are added to the LBBD master problem:

$$Q_{hd} \ge \sum_{p\in\mathcal{P}} \bar{\gamma}_p x_{hdp} + \bar{\beta} y_{hd} + \sum_{r\in\mathcal{R}_h} \bar{\delta}_r$$

where $\bar{\gamma}_p$, $\bar{\beta}$, and $\bar{\delta}_r$ are the optimal solutions of their corresponding dual variables.

For the decomposition of LBBD subproblem, we relax the variable z_{pr}^s as a continuous variable in subproblem (18). We denote the relaxed subproblem by $\theta_{sr}^{\mathrm{LP}}(\check{x}_{\cdot r}, T_{\cdot}^s)$, and shorten it as $\ddot{\theta}_{sr}^{\mathrm{LP}}$ later in the text:

$$\theta_{sr}^{\mathrm{LP}}(\check{x}_{\cdot r}, T_{\cdot}^s) = \min \sum_{p\in\mathcal{P}} -c_p z_{pr}^s$$

$$\text{s.t.} \sum_{p\in\mathcal{P}} T_p^s z_{pr}^s \le B_{hd} \qquad (\eta)$$

$$z_{pr}^s \le \check{x}_{pr} \qquad \forall p \in \mathcal{P} \qquad (\iota)$$

$$z_{pr}^s \in \{0,1\} \qquad \forall p \in \mathcal{P}$$

The corresponding classical Benders cut is:

$$\theta_{sr} \ge \sum_{p\in\mathcal{P}} \ddot{\imath}_p x_{pr} + B_{hd} \ddot{\eta}$$

where $\ddot{\eta}$ and $\ddot{\imath}_p$ are the optimal values for the corresponding dual variables.

D Overall Implementation Approach for Three-stage Decomposition

In this section we first describe the LBBD decomposition, then explain the decomposition of LBBD subproblem.

LBBD decomposition:

Phase one: This phase is very similar to the phase one of two-stage decomposition in Section 5.6. We first use the adapted FFD heuristic to obtain an initial solution. This solution is added as a warm start in the commercial solver to provide a feasible solution at the start of branch-and-cut. We also generate LBBD cuts and classical Benders cuts from this solution and add them to the

LBBD master problem. Next, we generated the constraints (20) from the subproblem relaxations and also add them to the LBBD master problem.

Phase two: We obtain the LBBD master problem with extra cuts and constraints from phase one and solve it with branch-and-cut. At each branch-and-bound node solve the node LP relaxation. If the objective value is greater or equal to the incumbent UB, then the current node can be pruned. Otherwise if the objective value is less than the incumbent UB, we proceed to check the integrality of $(\hat{u}, \hat{y}, \hat{x}, \hat{w})$ in the master solution. If $(\hat{u}, \hat{y}, \hat{x}, \hat{w})$ is integral, then solve the corresponding LBBD subproblems (15) with further decomposition (described with detail in the next paragraph). In the process of solving the LBBD subproblem, check if we need early stopping as explained in Section 6.3.2. If the solving process early stops, add LBBD cuts (19); otherwise, solve the LBBD subproblem to get the optimal objective value $\bar{Q}$. Also, solve subproblem LP relaxations and get optimal objective value $\bar{Q}^{\text{LP}}$. Decide if we can insert an additional heuristic solution as described in Section 5.5.3. Also, generate LBBD cuts and classical Benders cuts. In the CPLEX lazy constraint callback, compare the master solution of $\hat{Q}$ with $\bar{Q}$ and $\bar{Q}^{\text{LP}}$. If $\hat{Q} < \bar{Q}$ then add LBBD cuts; if $\hat{Q} < \bar{Q}^{\text{LP}}$ add classical Benders cuts. On the other hand, if some elements in the solution $(\hat{u}, \hat{y}, \hat{x}, \hat{w})$ are fractional, we only solve the subproblem LP relaxations, obtain $\bar{Q}^{\text{LP}}$, generate the classical Benders cuts and implement them if $\hat{Q} < \bar{Q}$ within the CPLEX user cut callback. We use the same user cut management as in Section 5.6 to manage those user cuts. After cutting planes are added in the CPLEX lazy constraint callback or the user cut callback, the node LP relaxation is solved again with those additional cutting planes. We repeat this process, until the stopping criteria is met, i.e. the gap between branch-and-bound UB and LB is small enough. In our implementation we stop the algorithm when such a gap is no more than 1%.

Decomposition of LBBD subproblem:

Phase one: Use the adapted FFD heuristic to obtain an initial solution. This solution is added as a warm start in the commercial solver to provide a feasible solution at the start of branch-and-cut.

Phase two: We solve the BDD master problem with branch-and-cut. At each branch-and-bound node solve the node LP relaxation. If the objective value is greater or equal to the incumbent UB, then the current node can be pruned. Otherwise if the objective value is less than the incumbent UB, we proceed to check the integrality of $\breve{x}$ in the master solution. If $\breve{x}$ is integral, then solve the corresponding BDD subproblems (18) and the BDD subproblem LP relaxations to get their respective optimal objective values $\ddot{\theta}$ and $\ddot{\theta}^{\text{LP}}$. Generate BDD-based Benders cuts and classical Benders cuts. In the CPLEX lazy constraint callback, compare the master solution of $\breve{\theta}$ with $\ddot{\theta}$ and $\ddot{\theta}^{\text{LP}}$. If $\breve{\theta} < \ddot{\theta}$ then add BDD-based Benders cuts; if $\breve{\theta} < \ddot{\theta}^{\text{LP}}$ add classical Benders cuts. On the other hand, if some elements in the solution $\breve{x}$ are fractional, we only solve the subproblem LP relaxations, obtain $\ddot{\theta}^{\text{LP}}$, generate the classical Benders cuts and implement them if $\breve{\theta} < \ddot{\theta}$ within the CPLEX user cut callback. The user cut management and stopping criteria are the same as in the LBBD decomposition.

E Parameter Values for Computational Analysis

Table A.1: Parameter values

κ_1	50 dollars
κ_2	-5 dollars
κ_3	-80 dollars
κ_4	-100 dollars
Γ	500
ρ_p	Uniform distribution in {1,2,...,5}, where 1 is the least urgent 5 is the most urgent
B_{hd}	Uniform distribution [420, 480] minutes in 15-minute intervals
α_p	Uniform distribution [60, 120] days
F_{hd}	Uniform distribution [4000, 6000]
G_{hd}	Uniform distribution [1500, 2500]
c_{dp}^{sched}	$\kappa_1 \rho_p (d - \alpha_p)$
c_p^{unsched}	$\kappa_2 \rho_p (\lvert\mathcal{D}\rvert + 1 - \alpha_p)$
c_p^{cancel}	$\kappa_3 \rho_p (\lvert\mathcal{D}\rvert + 1 - \alpha_p), \forall p \in \mathcal{P} \setminus \mathcal{P}'$
c_p^{cancel}	$\kappa_4 \rho_p (\lvert\mathcal{D}\rvert + 1 - \alpha_p), \forall p \in \mathcal{P}'$
T_p^s	Truncated lognormal distribution with mean 160 minutes, standard deviation 40, and truncation range [45, 480]
ω_p	$(\alpha_p - \lvert\mathcal{D}\rvert)\rho_p$

F Best Integer Solution Results for Algorithm Comparison

Table A.2: Comparison of Algorithms (continued): Best Integer Solution Objective Values (i.e., Upper Bounds)

instance (p-h-d-r)	MIP	2-BDD	2-LBBD	3-LBBD
10-2-3-3	-117624	-117624	-117670	-117670
25-2-3-3	-248582	-252010	-253491	-249238
10-3-5-3	-117227	-117671	-117595	-117671
25-3-5-3	-241882	-247676	-247807	-248107
50-3-5-3	-426982	-433702	-391874	-380486
75-3-5-3	-600867	-677288	-679017	-638701
10-2-3-5	-119481	-119551	-118935	-118935
25-2-3-5	-253762	-253335	-255321	-
50-2-3-5	-356952	-435996	-452404	-
75-2-3-5	-611189	-696665	-694526	-
10-3-5-5	-119555	-119537	-117611	-119588
25-3-5-5	-240490	-251945	-251931	-
50-3-5-5	-359391	-449867	-448060	-
75-3-5-5	-613869	-676269	-726670	-

Notice that for instance 10-2-3-5, both 2-BDD and 3-LBBD are solved to optimality, but they have different best integer results. This difference is caused by setting the 1% relative MIP gap in the solver.