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

Cheng Guo\textsuperscript{1}, Merve Bodur\textsuperscript{1}, Dionne M. Aleman\textsuperscript{1,2,3}, and David R. Urbach\textsuperscript{2,4,5}

\textsuperscript{1}Department of Mechanical and Industrial Engineering, University of Toronto, Toronto, Ontario M5S 3G8, Canada
\textsuperscript{2}Institute of Health Policy, Management and Evaluation, University of Toronto, Toronto, Ontario M5S 3E3, Canada
\textsuperscript{3}Techna Institute at University Health Network, Toronto, Ontario M5G 1P5, Canada
\textsuperscript{4}Department of Surgery, Women’s College Hospital, Toronto, Ontario M5S 1B2, Canada
\textsuperscript{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 (LBBDD) optimality cuts and binary decision diagram based Benders cuts. Specifically, to the best of our knowledge, deriving LBBDD 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 2018 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 (\textsuperscript{CIHI}, 2018). As a result of such lengthy waiting times for medically necessary treatments, patients suffer from lost wages and reduced productivity, due to the effects

\begin{itemize}
\item \texttt{cguo@mie.utoronto.ca}
\item \texttt{bodur@mie.utoronto.ca}
\item \texttt{aleman@mie.utoronto.ca}
\item \texttt{david.urbach@wchospital.ca}
\end{itemize}
of an untreated medical condition on mind and body (Barua and Jacques, 2019). The research on operating room (OR) scheduling studies how to allocate medical resources more strategically to improve the utilization of OR resources. In this work, we study the stochastic distributed OR scheduling problem (SDORS), 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 more uncertainties to the process, 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 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. 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. 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 allo-
cation of resources related to surgeries. In this work, we focus on both hospital and OR opening
decisions as well as surgery scheduling decisions.

The OR scheduling is well-studied in the deterministic setting, where we assume all parameters
are known. Jebali et al. (2006) consider an OR scheduling problem to assign operations to ORs.
Their goal is to minimize the summation of OR overtime, OR undertime, and the patient waiting
time. Marques et al. (2012) propose a mixed-integer programming (MIP) model that improves the
OR occupancy rate and shortens the surgery waiting time.

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 obtain cost savings (Min and Yih, 2010). Denton et al. (2010) minimize the cost in an OR
scheduling problem assuming stochastic overtime. Min and Yih (2010) consider a stochastic OR
scheduling problem with waiting list and downstream resource constraints. They model the problem
as a 2SIP and solve it with SAA. Deng et al. (2019) deal with uncertainty in both waiting times
and surgery durations. They use a distributionally robust chance constraint modeling approach in
the presence of ambiguous data.

The importance of distributed planning and scheduling gained attention in many industries in-
cluding 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 formula-
tions 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 a MIP model and an LBBD algorithm for
solving 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 collaborative environ-
ment. 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, 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 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. However, the L-Shaped methods assume that the master problem variables that are connected to the subproblem are all binary.

On the other hand, 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, thus it can handle mixed-integer subproblems. 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. However, those papers formulate only LBBD feasibility cuts for 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 consist 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 lead 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 hospitals 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.

Since 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.

Except the stochasticity of surgery durations and cancellation decisions, all other settings in the SDORS model are the same as in its deterministic counterpart. 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. Each patient in the set $\mathcal{P}$ has a health status score $\omega_p = (\alpha_p - |\mathcal{D}|)\rho_p$, where $\alpha_p$ is the number of days the patient has been waiting since her referral date, and $\rho_p$ is the patient’s urgency score. Given a health score threshold $\Gamma$, 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, $\mathcal{P}' = \{p \in \mathcal{P} \mid (\alpha_p - |\mathcal{D}|)\rho_p \geq \Gamma\}$ |
| $\mathcal{H}$            | Set of hospitals, $h \in \mathcal{H}$                          |
| $\mathcal{D}$            | Set of days in 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 $p$                                |
| $\rho_p$                 | Health status score assigned to patient $p$                     |
| $\alpha_p$               | The number of days elapsed since the referral date of patient $p$ |
| $\Gamma$                | Health status threshold above which patients have to be scheduled |
| $c_{sched}^p$           | Benefit for scheduled patient $p$ whose surgery is scheduled on day $d$ |
| $c_{unsched}^p$         | Penalty for unscheduled patient $p$                            |
| $c_{cancel}^p$          | Penalty for cancelled patient $p$                               |

| Decision variables:      |                                                                 |
|--------------------------|------------------------------------------------------------------|
| $u_{hd}$                 | 1 if the surgical suite in hospital $h$ is opened on day $d$, 0 otherwise |
The SDORS problem is formulated as follows:

\( \text{(SDORS)}: \min \sum_{h \in H} \sum_{d \in D} G_{hd} u_{hd} + \sum_{h \in H} \sum_{d \in D} \sum_{r \in R_h} F_{hd} y_{hdr} + \sum_{h \in H} \sum_{d \in D} \sum_{p \in P} \sum_{r \in R_h} c_{\text{sched}}^{x_{hdpr}} + \sum_{p \in P \setminus P'} c_{p}^{\text{unsched}} w_p + E T(x, y, T) \)  

s.t.  
\( \sum_{h \in H} \sum_{d \in D} \sum_{r \in R_h} x_{hdpr} = 1 \quad \forall p \in P' \)  
\( \sum_{h \in H} \sum_{d \in D} \sum_{r \in R_h} x_{hdpr} + w_p = 1 \quad \forall p \in P \setminus P' \)  
\( y_{hdr} \leq y_{hdr, r-1} \quad \forall h \in H, d \in D, r \in R_h \setminus \{1\} \)  
\( \sum_{p \in P} c_p^{\text{cancel}} x_{hdpr} \leq \sum_{p \in P} c_p^{\text{cancel}} x_{hdpr, r-1} \quad \forall h \in H, d \in D, r \in R_h \setminus \{1\} \)  
\( y_{hdr} \leq u_{hd} \quad \forall h \in H, d \in D, r \in R_h \)  
\( x_{hdpr} \leq y_{hdr} \quad \forall h \in H, d \in D, p \in P, r \in R_h \)  
\( u_{hd}, y_{hdr}, x_{hdpr}, w_p \in \{0, 1\} \quad \forall h \in H, d \in D, p \in P, r \in R_h \)  

where

\( Q(x, y, T) = \min \sum_{h \in H} \sum_{d \in D} \sum_{p \in P} \sum_{r \in R_h} c_p^{\text{cancel}} (x_{hdpr} - z_{hdpr}) \)  

s.t.  
\( \sum_{p \in P} T_{p} z_{hdpr} \leq B_{hd} y_{hdr} \quad \forall h \in H, d \in D, r \in R_h \)  
\( z_{hdpr} \leq x_{hdpr} \quad \forall h \in H, d \in D, p \in P, r \in R_h \)  
\( z_{hdpr} \in \{0, 1\} \quad \forall h \in H, d \in D, p \in P, r \in R_h \)

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

The first stage objective function (1a) minimizes the total operational and expected cancellation cost, whose terms are respectively (i) the total cost of opening 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 rooms 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 \((h,d)\). Constraints (1f) link \(u\) and \(y\) variables, to ensure that if 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 \(T\). To overcome this difficulty, we solve the problem approximately by SAA. We generate a set of scenarios for the surgery duration vector \(T\). The set of scenarios are denoted by \(S\) and the surgery duration of patient \(p\) under scenario \(s \in S\) is denoted by \(T^s_p\). Under each scenario we need to decide whether to operate or cancel a surgery, thus we now have a decision variable \(z^s_{hdp} \) for each scenario that represents such decisions. We assume that all scenarios are equally likely, so each has a realization probability of \(1/|S|\).

Replacing the expected cancellation cost with the average cancellation cost over the scenarios, we obtain the DE of the 2SIP:

\[
\text{(DE)}: \min \sum_{h \in H} \sum_{d \in D} G_{hd}u_{hd} + \sum_{h \in H} \sum_{d \in D} \sum_{r \in R_h} F_{hd}y_{hdr} + \sum_{h \in H} \sum_{d \in D} \sum_{p \in P} \sum_{r \in R_h} c_{dp}^{\text{assign}} x_{hdpr} + \sum_{p \in P \setminus P'} c_{p}^{\text{unsched}} w_p + \frac{1}{|S|} \sum_{s \in S} \sum_{h \in H} \sum_{d \in D} \sum_{p \in P} \sum_{r \in R_h} c_{p}^{\text{cancel}} (x_{hdpr} - z^s_{hdp})
\]  

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

\[
\sum_{p \in P} T^s_p z^s_{hdp} \leq B_{hd} y_{hdr} \quad \forall h \in H, d \in D, r \in R_h, s \in S \quad \text{(3b)}
\]

\[
z^s_{hdp} \leq x_{hdpr} \quad \forall h \in H, d \in D, p \in P, r \in R_h, s \in S \quad \text{(3c)}
\]

\[
z^s_{hdp} \in \{0, 1\} \quad \forall h \in H, d \in D, p \in P, r \in R_h, s \in S \quad \text{(3d)}
\]

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

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: classical Benders cuts from their BDD reformulations (BDD-based Benders cuts), LBBD optimality 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 \( Q_{s hdr} \) representing the cancellation cost of the hospital-day-OR combination \((h, d, r)\) under scenario \(s\):

\[
\begin{align*}
\min & \sum_{h \in H} \sum_{d \in D} G_{hd} u_{hd} + \sum_{h \in H} \sum_{d \in D} \sum_{r \in R} F_{hdr} y_{hdr} + \sum_{h \in H} \sum_{d \in D} \sum_{p \in P} \sum_{r \in R} c_{s hdr} x_{hdpr} \\
& + \sum_{p \in P \setminus P'} c_p^{unsched} w_p + \frac{1}{|S|} \sum_{s \in S} \sum_{h \in H} \sum_{d \in D} \sum_{r \in R} Q_{s hdr} \\
\text{s.t.} & \quad (1b) - (1h) \\
& \quad [LBBD cuts or BDD-based Benders cuts] \quad (4c) \\
& \quad Q_{s hdr} \geq 0, \quad \forall s \in S, h \in H, d \in D, r \in R_h \quad (4d)
\end{align*}
\]

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 \(Q_{s hdr}\). Denote 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 expected 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:

\[
\begin{align*}
\hat{Q}_{s hdr} &= \min \sum_{p \in P} c_p^{cancel}(\hat{x}_{hdpr} - \hat{z}_{hdpr}) \\
\text{s.t.} & \quad \sum_{p \in P} T_{p s hdr} \leq B_{hd} \quad (5b)
\end{align*}
\]
\[ z^s_{hdr} \leq \hat{x}_{hdr} \quad \forall p \in P \]  
\[ z^s_{hdr} \in \{0,1\} \quad \forall p \in P \]  

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.

### 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 results in a feasible subproblem, but wrongly estimated the subproblem optimal value, 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 \((\hat{u}_{hdr}, \hat{y}_{hdr}, \hat{x}_{hdr}, \hat{w}_p)\) from the master problem, while the subproblem objective \( \tilde{Q}^s_{hdr} \) provides the actual cancellation cost of such an assignment. When \( \hat{Q}^s_{hdr} < \tilde{Q}^s_{hdr} \), we add the following LBBD optimality cut to the LBBD master problem:

\[ Q^s_{hdr} \geq \tilde{Q}^s_{hdr} - \sum_{p \in \hat{P}_{hdr}} c^\text{cancel}_p (1 - x_{hdr}) \]  

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

Note that the LBBD cut is tight at the current master solution, i.e., it provides the exact cancellation cost for the master solution \((\hat{u}_{hdr}, \hat{y}_{hdr}, \hat{x}_{hdr}, \hat{w}_p)\). 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 a superset of \( \hat{P}_{hdr} \), or that contains a subset of \( \hat{P}_{hdr} \).

Theorem 1 states that the proposed LBBD cut eliminates the current master solution, and more importantly it does not exclude any optimal solution of (DE). Considering that there are finitely many binary \( x_{hdr} \) 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 formal proof for the validity of the LBBD cuts is given in the Appendix A.1.
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). Their method is applicable here because (i) in our decomposition the master variables that connect the subproblems, $x_{hdp}$'s, are binary variables; (ii) any subproblem constraint that is impacted by the first stage decisions is deactivated by a single $x_{hdp}$ variable.

The BDD-based Benders cuts are actually a set of classical Benders optimality cuts from the reformulation, which now is an LP.

In our problem, we transfer each subproblem to a BDD. The following discussion deals with the subproblem corresponding to an $(h, d, r, s)$ tuple and a master solution that schedules patients in $\hat{P}_{hdr}$. We only include patients from $\hat{P}_{hdr}$ in a BDD. Let $G_{hdr}^s = \{N_{hdr}^s, A_{hdr}^s\}$ be the BDD for this subproblem where $N_{hdr}^s$ is the set of nodes and $A_{hdr}^s$ is the set of arcs. Let $A_{hdr0}^s$ and $A_{hdr1}^s$ be the sets of zero-arcs and one-arcs, i.e., corresponding to setting variables in $\{z_{hdp}\}_{p \in \hat{P}_{hdr}}$ equal to 0 and 1 in the subproblem, respectively. In addition, we introduce the sets $A_{hdr0}^s \subset A_{hdr}^s$ and $A_{hdr1}^s \subset A_{hdr}^s$ to respectively denote the sets of zero-arcs and one-arcs corresponding to a patient $p$. Nodes $o \in N_{hdr}^s$ and $t \in N_{hdr}^s$ are the root and terminal nodes of the diagram $G_{hdr}^s$ and the capacity of all arcs are one. We denote the cost of an arc $a \in A_{hdr}^s$ as $g_{hdra}$ and its value is decided as follows:

$$g_{hdra} = \begin{cases} -c_{cancel}^p, & \text{if } a \in A_{hdr1}^s \\ 0, & \text{if } a \in A_{hdr0}^s \end{cases}$$

In Appendix B we provide a simple example to illustrate the process of the BDD reformulation.

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:

$$\tilde{Q}_{hdr}^s - \sum_{p \in P} c_p^{cancel} \tilde{x}_{hdp} = \min \sum_{a \in A_{hdr}^s} g_{hdra} f_a$$

s.t. \hspace{1cm} \sum_{a \in A_{hdr}^s, s(a) = o} f_a = 1 \hspace{1cm} (\pi_o) \hspace{1cm} (7a)

$$\sum_{a \in A_{hdr}^s, d(a) = i} f_a - \sum_{a \in A_{hdr}^s, d(a) = i} f_a = 0 \hspace{1cm} \forall i \in N^s \setminus \{o, t\} \hspace{1cm} (\pi_i) \hspace{1cm} (7b)$$

$$\sum_{a \in A_{hdr}^s, d(a) = t} f_a = -1 \hspace{1cm} (\pi_t) \hspace{1cm} (7c)$$

$$f_a \leq \tilde{x}_{hdp} \hspace{1cm} \forall a \in A_{hdr1}^s, p \in P \hspace{1cm} (\neg \xi_a) \hspace{1cm} (7d)$$

$$f_a \geq 0 \hspace{1cm} \forall a \in A_{hdr}^s \hspace{1cm} (7e)$$

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 (7) is different from that of the subproblem (5) by a constant term $\sum_{p \in P} c_p^{cancel} \tilde{x}_{hdp}$.

Denote the optimal dual solutions for $\pi_i$ and $\xi_a$ respectively as $\bar{\pi_i}$ and $\bar{\xi_a}$. We can derive the
following $BDD$-based Benders cut from the problem (7):

$$Q_{hdr}^s - \sum_{p \in P} c_p^{cancel} x_{hdpr} \geq \bar{\pi}_r - \sum_{p \in P} \left( \sum_{a \in A_{hdpr}^1} \xi_a \right) x_{hdpr}$$

(8)

As suggested by Lozano and Smith (2018), this cut can be further strengthened by replacing the $x_{hdpr}$ coefficient of $\sum_{a \in A_{hdpr}^1} \xi_a$, with $\max_{a \in A_{hdpr}^1} \xi_a$.

$$Q_{hdr}^s - \sum_{p \in P} c_p^{cancel} x_{hdpr} \geq \bar{\pi}_r - \sum_{p \in P} \left( \max_{a \in A_{hdpr}^1} \xi_a \right) x_{hdpr}$$

(9)

In our implementation we use this strengthened version of $BDD$-based Benders cut. For the proof of the validity of (9), 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 $Q_{hdr}^s$, 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 $Q_{hdr}^{s,LP}$. Let $\delta$ and $\eta$ denote the optimal dual solutions that respectively correspond to constraints (5b) and (5c) in the LP relaxation. The classical Benders cut (10), that is generated and added to the master problem (4) when $\hat{Q}_{hdr}^s < Q_{hdr}^{s,LP}$, is as follows:

$$Q_{hdr}^s \geq \sum_{p \in P_{hdr}} (c_p^{cancel} + \bar{\delta}_p)x_{hdpr} + \bar{\eta}B_{hd}$$

(10)

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 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 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 \(\omega_p\).

Overall, the heuristic has the following steps. (i) Randomly pick a scenario \(s \in S\) (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 under 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 conclude that the adapted FFD heuristic cannot fit all the 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.

Based on the heuristic solution, we derive LBBB cuts (6) or BDD-based Benders cuts, plus classical Benders cuts (10). 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:

\[
Q^s_{hdr} \geq \left( \min_{p \in P} \frac{\epsilon_p^{cancel}}{T_s} \right) \left( \sum_{p \in P} T_s x_{hdp} - B_{hd} \right) \quad \forall h \in H, d \in D, r \in R_h, s \in S \tag{11}
\]

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 Appendix A.2.

**Theorem 2.** Constraints (11) provide valid LBs for \(Q^s_{hdr}\).

### 5.5.3 Insert Heuristic Solution in Branch-and-Cut

Inspired by the practice in Bodur and Luedtke (2016), we add additional solutions through the commercial solver callback inside the branch-and-cut algorithm. An 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 \(\hat{Q}\), then the solution \((\hat{u}, \hat{y}, \hat{x}, \hat{w}, \hat{Q})\) is a feasible solution for \((DE)\). 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 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.

5.6 Overall Implementation Approach

As mentioned before, we add the cutting planes, including the LBBD cuts (6), the BDD-based Benders cuts (9), and the classical Benders cuts (10), through branch-and-cut. This is different from the standard cutting plane implementation, where master problem need to be solved again each time new cutting planes are generated.

The overall implementation of the two-stage decomposition algorithm is illustrated as a flow chart in Appendix C. 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 at the start of branch-and-cut. 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 (11) 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 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 solve the corresponding subproblems (5) and subproblem LP relaxations, and get their respective optimal objective values \(\bar{Q}\) and \(\bar{Q}_{LP}\). Decide if we can update the incumbent UB with a heuristic solution as described in Section 5.5.3. Also, 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 \(\bar{Q}_{LP}\). If \(\hat{Q} < \bar{Q}\), add LBBD cuts or BDD-based Benders cuts; if \(\hat{Q} < \bar{Q}_{LP}\) 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 \(\bar{Q}_{LP}\), generate the classical Benders cuts and implement them if \(\hat{Q} < \bar{Q}\) within the CPLEX user cut callback, which we refer to 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 develop a two-stage decomposition for the subproblems.

In Section 6.1, we introduce the decomposition for the (DE) and the LBBD cuts. In Section 6.2, we show how the computationally expensive subproblems can be further decomposed via the BDD-based method. Additional algorithmic enhancements, some of which 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 \textit{LBBD master problem} and \textit{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 \(w_p\), as well as new variables \(y_{hd}\), \(x_{hdp}\) and \(Q_{hd}\). 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 \(r \in R_h\) 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 patients assigned to hospital \(h\) on day \(d\). Also we use a new variable, \(Q_{hd}\), to denote the \textit{expected} cancellation cost for an \((h, d)\) pair:

\[
\begin{aligned}
\min & \quad \sum_{h \in H} \sum_{d \in D} G_{hd} u_{hd} + \sum_{h \in H} \sum_{d \in D} F_{hd} y_{hd} + \sum_{h \in H} \sum_{d \in D} \sum_{p \in P} c_{p}^{\text{sched}} x_{hdp} \\
& \quad + \sum_{p \in P \setminus P'} c_{p}^{\text{unsched}} w_p + \sum_{h \in H} \sum_{d \in D} Q_{hd} \\
\text{s.t.} & \quad \sum_{h \in H} \sum_{d \in D} x_{hdp} = 1 & \forall p \in P' \\
& \quad \sum_{h \in H} \sum_{d \in D} x_{hdp} + w_p = 1 & \forall p \in P \setminus P' \\
& \quad y_{hd} \leq |R_h| u_{hd} & \forall h \in H, d \in D 
\end{aligned}
\]  

(12a)
Constraints (12b) and (12c) assign patients to hospitals, as well as to days of the current planning horizon or a future planning horizon. Constraints (12d) ensure that when a hospital \( h \) is not opened on a day \( d \), there is no OR in that hospital to be opened. Constraints (12e) make sure that when a hospital \( h \) has a patient assigned on day \( d \), there must be at least one room opened in that hospital. Those constraints are not necessary for a correct formulation, but they make the formulation tighter. Constraints (12f) is a set of 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 open the given number of ORs and 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 optimal solutions of \( x_{hdp} \) and \( y_{hd} \) to the subproblem. The subproblem for \((h, d)\) is as below:

\[
\bar{Q}_{hd} = \min \frac{1}{|S|} \sum_{s \in S} \sum_{p \in P} \sum_{r \in R_h} c_p^{\text{cancel}} (x_{pr} - z_{sp}^s) \\
\text{s.t.} \\
\sum_{r \in R_h} x_{pr} = \hat{x}_{hdp} \quad \forall p \in P \\
\sum_{p \in P} T_{ps}^s x_{pr} \leq B_{hd} \quad \forall r \in R_h, s \in S \\
z_{sp}^s \leq x_{pr} \quad \forall p \in P, r \in R_h, s \in S \\
x_{pr} = 0 \quad \forall r \geq \hat{y}_{hd} + 1 \\
x_{pr}, z_{sp}^s \in \{0, 1\} \quad \forall p \in P, r \in R_h, s \in S
\]

Constraints (13b) 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 (13c) specify time limit. Constraints (13d) ensure that if a patient is not scheduled then she will not be operated on. Constraints (13e) are symmetry breaking constraints for ORs. Those constraints ensure that for a hospital-day pair, rooms with lower indices are opened before rooms 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. However, since the LBBD master problem is a relaxation of the original problem, (DE), $\tilde{Q}_{hd}$ is a LB on the actual optimal cancellation cost corresponding to $(\tilde{u}_{hd}, \tilde{y}_{hd}, \tilde{x}_{hdp}, \tilde{w}_p)$. When $\tilde{Q}_{hd} < \bar{Q}_{hd}$, we add the followings cuts, which we refer to as the LBBD optimality cut, to the LBBD master problem:

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

\[
y_{hd} \geq (1 + \hat{y}_{hd})(1 - g_{hdj})
\]

\[
g_{hdj} \in \{0, 1\}
\]

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

The cut is deducted from the logic $y_{hd} \leq \hat{y}_{hd} \Rightarrow Q_{hd} \geq \bar{Q}_{hd} - \sum_{p \in \hat{P}_{hd}} \bar{Q}_{hd}(1 - x_{hdp})$, which means the LB on $Q_{hd}$ is imposed only when $y_{hd} \leq \hat{y}_{hd}$. 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 Appendix A.3.

**Theorem 3.** The LBBD optimality cuts (14) are valid.

6.2 Decomposition of the LBBD Subproblems

The LBBD subproblem (5) is itself a 2SIP. The first stage decision is to assign the patients in the list $\hat{P}_{hd}$ to ORs, which corresponds to the decision variables $x_{pr}$, the recourse option is the acceptance decision under each scenario, represented by the decision variables $z_{ps}$. Solving such a 2SIP can be time consuming. For example in our experiment, for an instance with $|S| = 50, |P| = 20, |H| = 3, |R_h| = 3$, 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 variable $\theta_{sr}$ corresponds to the optimal objective value of the subproblem for OR $r$ in scenario $s$. The BDD master problem is as follows:

\[
\hat{Q}_{hd} = \min \sum_{p \in P} \sum_{r \in R_h} c_p^{cancel} x_{pr} + \frac{1}{|S|} \sum_{s \in S} \sum_{r \in R_h} \theta_{sr} 
\]

\[
s.t. \sum_{r \in R_h} x_{pr} = \hat{x}_{hdp} \quad \forall p \in P
\]
\[ x_{pr} = 0 \quad \forall r \geq \hat{y}_{hd} + 1 \]  
\[ \sum_{p \in P} c_p^{\text{cancel}} x_{pr} \leq \theta_{sr} \leq 0 \quad \forall s \in S, r \in R_h \]  
[BDD-based Benders cuts]  
\[ x_{pr} \in \{0, 1\} \quad \forall p \in P \]

The LHS of the equation (15d) is derived from the relaxation of the subproblem (without the time limit constraint). Constraints (15e) are a set of BDD-based Benders cuts generated from the transformed BDD subproblem that we will define below. All the other constraints have the same meaning as their counterparts in the subproblem (13).

We use \( \bar{x}_{pr} \) 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:

\[ \bar{\theta}_{sr} = \min \sum_{p \in P} -c_p^{\text{cancel}} z_{pr}^s \]  
\[ \text{s.t.} \sum_{p \in P} T_s^p z_{pr}^s \leq B_{hd} \]  
\[ z_{pr}^s \leq \bar{x}_{pr} \quad \forall p \in P \]  
\[ z_{pr}^s \in \{0, 1\} \quad \forall p \in P \]

where the optimal objective value of the BDD subproblem is denoted by \( \bar{\theta}_{sr} \). Constraints (16b) and (16c) are respectively the time limit constraints and the constraints for UB of \( z_{pr}^s \).

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 (9).

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 sub-optimal 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 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 \( \theta_{sr} \). The procedure of deriving
those classical Benders cuts is standard. We describe the process and the classical Benders cuts in Appendix D

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. This 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 \(\text{globalUB}\) and the operational cost at the current integral solution as \(\text{incumbentOptCost}\). When solving the LBBD subproblem at such a node, we record the best LB of the subproblem whenever one is found, as \(\bar{Q}_{hd}^{LB}\). If \(\text{globalUB} < \text{incumbentOptCost} + \bar{Q}_{hd}^{LB}\), 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 useful to derive an LBBD cut from its best-known LB, \(\bar{Q}_{hd}^{LB}\). Such a cut is useful to cut off the current master solution, and has the same logic as (14), but we need to replace \(Q_{hd}\) in (14) with \(\bar{Q}_{hd}^{LB}\):

\[
\begin{align*}
Q_{hd} & \geq \bar{Q}_{hd}^{LB} \left( g_{hdj} - \sum_{p \in \hat{P}_{hd}} (1 - x_{hdp}) \right) \\
y_{hd} & \geq (1 + \hat{y}_{hd})(1 - g_{hdj}) \\
g_{hdj} & \in \{0, 1\}
\end{align*}
\]

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 \(\hat{P}_{hd}\) to it, the cancellation cost is at least \(\bar{Q}_{hd}^{LB}\).

6.3.3 Adapted FFD Heuristic

We use the adapted FFD heuristic to get an initial solution for the LBBD master problem as describe in Section 5.5.1. We also develop an adapted FFD heuristic for the BDD master problem.
to assign patients in $\hat{P}_{hd}$ to the $\hat{y}_{hd}$ opened ORs, and leave all patients that cannot be fitted to the one of the 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 $\hat{P}_{hd}$ in the decreasing order of cancellation cost, and assign patients to ORs in their sorted orders according to the surgery duration in their first scenario. For each patient we try to fit her into one of the ORs that already has patients in it. If none of the opened ORs can fit her, we use an OR that has not been used. 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 $\hat{\theta}_{sr}$.

Before the start of the branch-and-cut process, we add LBBD cuts (14) and classical Benders cuts based on the heuristic solution to the LBBD master problem (12).

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 $Q_{hd}$:

$$Q_{hd} \geq \frac{1}{|S|} \sum_{s \in S} \left( \min_{p \in P} \frac{c_{cancel}^{p}}{T_{s}} \right) \left( \sum_{p \in P} T_{s}^{p} x_{hdp} - B_{hd} \hat{y}_{hd} \right) \quad \forall h \in H, d \in D \tag{18}$$

and those constraints are valid:

**Theorem 4.** The constraints (18) are valid for (DE).

The proof is provided in Appendix A.4.

6.3.5 Overall Implementation

Similar to the two-stage decomposition, we employ a branch-and-cut approach, in both the LBBD decomposition and the decomposition of LBBD subproblem. More details on the implementation is provided in Appendix E.

7 Computational Results

We present the computational analysis of the SDORS model and decomposition algorithms.

In our analysis we use the same dataset and parameters from Roshanaei et al. (2017) wherever those data are available. The only difference from their dataset lies in the surgery durations and the cost parameter $c_{sched}^{dp}$, $c_{unsched}^{dp}$ and $c_{cancel}^{p}$, 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 surgery duration distribution come from a real surgery dataset of UHN.
The benefit of scheduled patients, \( e_{dp}^{\text{sched}} \), the penalty for unscheduled patients, \( e_{dp}^{\text{unsched}} \), and the penalty for cancelled patient, \( e_{p}^{\text{cancel}} \), are calculated as follows: let \( \kappa_1 \) be the unit benefit of scheduling patients on time, \( \kappa_2 \) be the unit penalty of postponing patients to future planning horizons, while \( \kappa_3 \) and \( \kappa_4 \) 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 \( e_{dp}^{\text{sched}} = \kappa_1 \rho_p(d - \alpha_p) \), \( e_{p}^{\text{unsched}} = \kappa_2 \rho_p(|D|+1 - \alpha_p) \), \( e_{p}^{\text{cancel}} = \kappa_3 \rho_p(|D|+1 - \alpha_p) \) for non-mandatory patients and \( e_{p}^{\text{cancel}} = \kappa_4 \rho_p(|D|+1 - \alpha_p) \) for mandatory patients.

The values of the parameters are provided in Table 6 of the Appendix.

### 7.1 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 our original two-stage stochastic program in a more concise way:

\[
\min_{X \in \mathcal{X}} f(X) = c^T X + \mathbb{E}_{\xi}[Q(X, \xi)]
\]  

(19)

where the set \( \mathcal{X} \) include all feasible first-stage variables \( X = \{ u_{hd}, y_{hd}, x_{hdp}, w_p \} \), \( c \) is their objective coefficients, \( \xi \) is the vector of random variables, and \( Q \) is the second stage value function. In our setting we assume \( \xi \) follows a truncated normal distribution.

Considering that we use \(|S|\) samples in (DE), in order to obtain a LB on the optimal value of problem (19) via SAA, we first generate \( N^{\text{Sample}} \) samples where each sample contains \(|S|\) scenarios of simulated surgery durations. Due to the limit of computational resource, \( N^{\text{Sample}} \) 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 value as it will be useful when calculating the UB. We denote the optimal value by \( F_n^{\text{LB}} \). We can then construct a 95% confidence interval (CI) for the LB of (19) from the \( N^{\text{Sample}} \) optimal objective values, \( \{F_n^{\text{LB}}\}_{n=1,...,N^{\text{Sample}}} \).

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 (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 \( S^{\text{Select}} \). Then for each solution \( X_n \), calculate the second-stage cancelation cost under each scenario \( s \in S \), denoted by \( Q(X_n, \xi_s) \). Choose the \( X_n \) with the smallest value of \( c^T X_n + \frac{1}{|S^{\text{Select}}|} \sum_{s \in S^{\text{Select}}} Q(X_n, \xi_s) \) and denote it by \( X^{\text{min}} \).

After fixing the first-stage solution to \( X^{\text{min}} \), we generate a set of scenarios, \( S^{\text{UB}} \). The cardinality of this set should be large. Then for each scenario \( s \) we evaluate the cancellation cost and denote it by \( Q(X^{\text{min}}, \xi_s) \). Since \( X^{\text{min}} \in \mathcal{X} \), the optimal objective value from each scenario, \( c^T X^{\text{min}} + Q(X^{\text{min}}, \xi_s) \), 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 \(|P| = 12, |H| = 2, |D| = 2, |R_h| = 2\) to conduct the SAA analysis. As the parameters of the SAA analysis, we choose \( N^{\text{Sample}} = 30, |S^{\text{Select}}| = 1,000 \) and \(|S^{\text{UB}}| = 10,000\).
We compare the worst case optimality gap, obtained as \(((\text{Mean of UB} + \text{Width of UB}) - (\text{Mean of LB} - \text{Width of LB})) / (\text{Mean of LB} - \text{Width of LB})\), when \(|S| = 25, 50, 75, 100\). 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 |
|---|---|---|---|
| | 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 fix the number of scenarios \(|S| = 100\) in the experiments for the following sections.

7.2 Algorithmic Comparisons

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. The computer used to perform those experiments runs MacOS and has a 2.7 GHz Intel Core i5 processor and a 16GB RAM. The time limit for these experiments is 30 minutes 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. 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 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.

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 MIP in all test instances. Especially for the instances with more patients, directly using the MIP results in gaps beyond 60%, while decomposition algorithms can solve those instances to a gap around 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 7.1 we used this method to solve all the optimization problems to optimality, since the SAA analysis uses an 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, using BDD cuts has a small advantage over using LBBD cuts, though the latter ties or even outperforms the former in certain instances. For the larger instances, the three-stage decomposition becomes very ineffective, as the size of the LBBD subproblems become the bottleneck.

### 7.3 Value of Incorporating Stochasticity

We evaluate the value of incorporating stochastic surgery durations in the collaborative OR scheduling problem by comparing the optimal schedule from the deterministic problem (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 7.2, 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 30 minutes and the optimality gap being set to 1%. Note that after 30 minutes, 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 of 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 sum up those evaluations from the 10,000 samples and take their averages.

Table 4: Comparison of the Deterministic and Stochastic Models

| instance (p-h-d-r) | Cancellation Rate (%) | Utilization Rate (%) |
|--------------------|-----------------------|----------------------|
|                    | Deterministic         | Stochastic           | Deterministic | Stochastic |
| 10-3-5-3           | 18.12                 | 0.64                 | 5.56         | 7.67       |
| 25-3-5-3           | 19.69                 | 6.26                 | 15.08        | 17.95      |
| 50-3-5-3           | 17.31                 | 15.47                | 31.20        | 32.05      |
| 75-3-5-3           | 15.65                 | 6.72                 | 47.94        | 53.67      |
| 10-3-5-5           | 18.16                 | 0.64                 | 3.33         | 4.60       |
| 25-3-5-5           | 19.36                 | 2.25                 | 9.09         | 11.31      |
| 50-3-5-5           | 16.82                 | 2.23                 | 18.46        | 22.63      |
| 75-3-5-5           | 14.94                 | 4.69                 | 29.02        | 32.98      |

In Table 4 we report the average cancellation rate and the average 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, especially when there are more patients and ORs, e.g., the instances 50-3-5-5 and 75-3-5-5.

Notice that all the deterministic models are solved to optimality, while most of the stochastic models have an optimality gap of around 20% when 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.

7.4 Sensitivity Analysis

In this section we perform several sensitivity analysis for the SDORS problem. 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 parameters or different optimality gaps. Therefore, all experiments in this section are conducted on an instance with $|\mathcal{P}| = 12, |\mathcal{H}| = 2, |\mathcal{D}| = 2, |\mathcal{R}_h| = 2$. This instance can be solved to optimality within a reasonable time (using the three stage decomposition), but also non-trivial enough to
produce interesting results.

We first solve the SDORS problem 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 $\kappa_3$ and $\kappa_4$ 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 7.3 and obtain average cancellation rates and average OR utilization rates. Results are shown in Table 5. We compare the baseline case where none of the parameters are changed, to the above three cases.

| Parameter Setting | Cancellation Rate (%) | Utilization Rate (%) |
|-------------------|-----------------------|----------------------|
| Baseline          | 8.08                  | 12.63                |
| Case 1            | 3.19                  | 13.22                |
| Case 2            | 16.73                 | 11.31                |
| Case 3            | 9.23                  | 12.21                |

From the results, we can see that for Case 1 where the standard deviation of surgery duration becomes higher, we have a lower cancellation rate and a higher OR utilization rate. Due to the higher variance of surgery durations, the model opens more ORs to allow more slack in the optimal schedule, which leads to less cancellations and a higher utilization rate in this test case.

For both Case 2 and Case 3, the cancellation rates increase and the OR utilization rate drops compared to the baseline case. Those results demonstrate that if it is less costly to cancel patients, then more patients will get cancelled and ORs will get used less. Also, if the opening hours of ORs become shorter, we will need to cancel more patients.

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 distributed bin packing problem, as the SDORS problem can be viewed as an extension of a 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] B. Barua and D. Jacques. The private cost of public queues for medically necessary care. *Fraser Research Bulletin*, 2019.

[3] 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.

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

[5] 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.

[6] CIHI. Wait times for priority procedures in Canada. *Canadian Institute for Health Information Ottawa, ON*, 2018.

[7] 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.

[8] Y. Deng, S. Shen, and B. Denton. Chance-constrained surgery planning under conditions of limited and ambiguous data. *INFORMS J. Comput.*, 2019.

[9] 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.

[10] 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.

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

[12] A. Jebali, A. B. H. Alouane, and P. Ladet. Operating rooms scheduling. *International J. Production Economics*, 99(1-2):52–62, 2006.
[13] 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.

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

[15] 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.

[16] 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.

[17] 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.

[18] 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.

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

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

[21] 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.

[22] 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.

[23] 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.

[24] 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.

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

[26] 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

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 \((\hat{u}_{hd}, \hat{y}_{hdr}, \hat{x}_{hdr}, \hat{w}_p, \hat{Q}_{hdr})\) violates the cut, i.e., it will be cut off. When we substitute the master solution to the cut (6), the left-hand side (LHS) of the cut becomes \( \hat{Q}_{hdr} \), while the right-hand side (RHS) of the cut becomes \( \bar{Q}_{hdr} \) as all \( \hat{x}_{hdr} \)’s in the set \( \hat{P}_{hdr} \) have the value 1. This gives us \( \hat{Q}_{hdr} \geq \bar{Q}_{hdr} \). However, we know that \( \hat{Q}_{hdr} < \bar{Q}_{hdr} \) 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 \((\hat{u}_h^*, \hat{y}_h^*, \hat{x}_{hdr}, \hat{w}_p, \bar{Q}_{hdr}^*)\), is not excluded by the LBBD optimality cut that is generated from a master solution \((\hat{u}_{hd}, \hat{y}_{hdr}, \hat{x}_{hdr}, \hat{w}_p, \hat{Q}_{hdr})\) using \( \hat{P}_{hdr} \). Here, \((\hat{u}_h^*, \hat{y}_h^*, \hat{x}_{hdr}, \hat{w}_p, \bar{Q}_{hdr}^*)\) is obtained by solving the master problem, and \( \bar{Q}_{hdr}^* \) is the corresponding optimal objective value of the subproblem. For a globally optimal solution, the cancellation cost obtained by the master problem, \( \bar{Q}_{hdr}^* \), should match that from the subproblem, \( \bar{Q}_{hdr}^s \).

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 \( \hat{x}_{hdr} = 0, \forall p \in P \) due to (1g). This in turn yields all \( \hat{z}_p^* = 0 \) in the optimal subproblem solution due to (5c) and thus \( \bar{Q}_{hdr}^s = 0 \). Replacing those values in the cut, we get the following:

\[
\bar{Q}_{hdr}^s \geq \bar{Q}_{hdr}^* - \sum_{p \in \hat{P}_{hdr}} c_p^{cancel} (1 - \hat{x}_{hdr}^*)
\]

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

\[
\bar{Q}_{hdr}^* \leq \sum_{p \in \hat{P}_{hdr}} c_p^{cancel}
\]

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 (\(*\)) is nonpositive. Therefore, the global optimal solution \((\hat{u}_h^*, \hat{y}_h^*, \hat{x}_{hdr}^*, \hat{w}_p^*, \bar{Q}_{hdr}^s)\) 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, \( \hat{x}_{hdr}^* \), we either still
have all the patients in the set \( \hat{P}_{hdr} \) 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 \( \hat{P}_{hdr} \) are assigned to the current \((h, d, r)\) tuple at an optimal solution, i.e., \( \hat{x}^*_{hdr} = 1, \forall p \in \hat{P}_{hdr} \), then the corresponding cancellation cost for the current patient list, \( \hat{Q}^s_{hdr} \), should not be lower than the cancellation cost for \( \hat{P}_{hdr} \), which is \( \hat{Q}^s_{hdr} \). That is, we have \( \hat{Q}^s_{hdr} \geq \hat{Q}^s_{hdr} \). Substitute the optimal solution into the LBBD cut:

\[
\hat{Q}^s_{hdr} \geq \hat{Q}^s_{hdr} - \sum_{p \in \hat{P}_{hdr}} c^\text{cancel}_p (1 - \hat{x}^*_{hdr}) = 0
\]

which holds.

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

To help illustrate the relationships between patient sets, we use the following simple example in Figure 1. 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., \( \{p_1, p_2, p_3\} \in \hat{P}_{hdr} \). In the figure they are colored with gray. The global optimal solution schedules the patient set \( P^{A*}_{hdr} = \{p_2, p_3, p_4, p_5\} \). Then according to our definition, \( \hat{P}^{A*}_{hdr} = P^{A*}_{hdr} \cap \hat{P}_{hdr} = \{p_2, p_3\} \), \( \hat{P}^{A*}_{hdr} = P^{A*}_{hdr} \setminus \hat{P}^{A*}_{hdr} = \{p_4, p_5\} \), and \( \hat{P}^{N*}_{hdr} = \hat{P}_{hdr} \setminus \hat{P}^{A*}_{hdr} = \{p_1\} \).

![Figure 1: Illustration of relationships between patient sets](image)

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

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

\[
\bar{Q}_{hdr}^* \geq \bar{Q}_{hdr}^i (\hat{P}_{hdr}^A) \geq \bar{Q}_{hdr} - \sum_{p \in \hat{P}_{hdr}^N} c_p^{cancel} (1 - \hat{x}_{hdrp}) - \sum_{p \in \hat{P}_{hdr}^A} c_p^{cancel} (1 - \hat{x}_{hdrp})
\]

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

At optimality \( Q_{hdr}^* \) equals the optimal objective value of the subproblem (5). We want to prove that the RHS of inequality (11) 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 \( Q_{hdr}^* \) is not opened. In this case, there should be no cost for cancelling as no patient is assigned in the first place. Therefore, \( Q_{hdr}^* = 0 \). Since in this case the RHS of the inequality becomes \( \min_{p \in P} \left( \frac{c_p^{cancel}}{T_p} \right) (B_{hdr}) < 0 \) as the consequence of \( x_{hdrp} = 0 \) (\( \forall p \in \mathcal{P} \)), the constraint is valid.

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

\[
\sum_{p \in \hat{P}_{hdr}^C} c_p^{cancel} = \sum_{p \in \hat{P}_{hdr}^C} c_p^{cancel} \hat{x}_{hdrp} \\
= \sum_{p \in \hat{P}_{hdr}^C} c_p^{cancel} \frac{T_p^{s}}{T_p} \hat{x}_{hdrp} \\
\geq \left( \min_{p \in P} \frac{c_p^{cancel}}{T_p} \right) \sum_{p \in \hat{P}_{hdr}^C} T_p^{s} \hat{x}_{hdrp} \\
\geq \left( \min_{p \in P} \frac{c_p^{cancel}}{T_p} \right) \left( \sum_{p \in \hat{P}_{hdr}^C} T_p^{s} \hat{x}_{hdrp} + \sum_{p \in \hat{P}_{hdr} \setminus \hat{P}_{hdr}^C} T_p^{s} \hat{x}_{hdrp} - B_{hdr} \right) \\
= \left( \min_{p \in P} \frac{c_p^{cancel}}{T_p} \right) \left( \sum_{p \in \hat{P}_{hdr}} T_p^{s} \hat{x}_{hdrp} - B_{hdr} \right)
\]
Therefore, for any assignment \( \hat{x}_{hdp} (\forall p \in P) \), the expression 
\[
\left( \min_{p \in P} c_{p}^{\text{cancel}} \right) \left( \sum_{p \in P} T_{p} \hat{x}_{hdp} - B_{hd} \right)
\]
provides a LB for \( Q_{hdr}^{\prime} \). Substitute the fixed assignment \( \hat{x}_{hdp} \) with the variable \( x_{hdp} \) and we get the constraint (11).

### A.3 Proof of Theorem 3

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

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

**Case 2**: If \( \hat{u}_{hd}^{\prime} = 1 \), then we discuss the following two cases separately:

**Subcase a**: If \( \hat{y}_{hd}^{\prime} > \hat{y}_{hd} \), 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 \( \hat{Q}_{hd} \). This is why we make the cut redundant in this case: let \( g_{hdj} = 0 \), (14a) becomes

\[
Q_{hd} \geq \hat{Q}_{hd} \left( - \sum_{p \in \hat{P}_{hd}} (1 - \hat{x}_{hdp}) \right)
\]

This is always true since RHS is nonpositive, so \( \hat{Q}_{hd}^{\prime} \) and \( \hat{x}_{hdp}^{\prime} \) also satisfy this inequality. (14b) 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 \( \hat{y}_{hd}^{\prime} \leq \hat{y}_{hd} \), then there are further two subcases to discuss. Note that due to (14b), 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. \( \hat{x}_{hdp} \geq \hat{x}_{hdp}^{\prime}, \forall p \in \hat{P}_{hd} \), then we claim that the optimal cancellation cost will not decrease from the current value \( \hat{Q}_{hd} \) 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 \( Q_{hd} < \hat{Q}_{hd} \) 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 \( \hat{Q}_{hd} \), which is a contradiction. Therefore, \( Q_{hd} \geq \hat{Q}_{hd} \). Let \( g_{hdj} = 1 \), and also replace \( Q_{hd} \) and \( x_{hdp} \) with \( \hat{Q}_{hd}^{\prime} \) and \( \hat{x}_{hdp}^{\prime} \), (14) becomes:

\[
\begin{align*}
\hat{Q}_{hd} & \geq Q_{hd} \\
\hat{Q}_{hd}^{\prime} & \geq Q_{hd} \left( g_{hdj} - \sum_{p \in \hat{P}_{hd}} (1 - \hat{x}_{hdp}) \right) \\
\hat{y}_{hd} & \geq (1 + \hat{y}_{hd})(1 - g_{hdj})
\end{align*}
\]
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 $Q'_{hd}$, because the cancellation cost can either be zero or be a nonzero value that is larger or smaller than $Q_{hd}$. Thus we make the LBB cut redundant:

$$
\begin{align*}
\bar{Q}_{hd} \geq 0 & \quad \left( \frac{1}{\gamma_{hdj}} - \sum_{p \in P_{hd}} (1 - \hat{x}'_{hdp}) \right) \\
\leq \gamma_{hd} & \quad \left( 1 + \hat{y}_{hd} \right) \left( 1 - \frac{1}{\gamma_{hdj}} \right)
\end{align*}
$$

which is satisfied by the global optimal solution.

A.4 Proof of Theorem 4

At optimality $Q_{hd}$ equals the optimal objective value of the subproblem (13). We want to prove that the RHS of inequality (18) 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 $x_{hdp} = 0$ ($\forall p \in P$) 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, $\hat{x}_{hdp}$ ($\forall p \in P$), the set of assigned patients, $\hat{P}_{hd}$, and the number of opened ORs, $\hat{y}_{hd}$. Suppose under the scenario $s \in S$, a set of patients, $\hat{P}^{sC}_{hdr} \subseteq \hat{P}_{hd}$, is cancelled as dictated by the optimal solution of the subproblem (13). Then by definition we have $Q_{hd} = \frac{1}{|S|} \sum_{s \in S} \sum_{p \in \hat{P}^{sC}_{hdr}} c_{p}^{\text{cancel}}$. 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 $s \in S$:

$$
\begin{align*}
\sum_{p \in \hat{P}^{sC}_{hdr}} c_{p}^{\text{cancel}} \hat{x}_{hdp} &= \sum_{p \in \hat{P}^{sC}_{hdr}} c_{p}^{\text{cancel}} \hat{x}_{hdp} \\
&= \sum_{p \in \hat{P}^{sC}_{hdr}} c_{p}^{\text{cancel}} T_{p}^{s} \hat{x}_{hdp} \\
&\geq \left( \min_{p \in \hat{P}} \frac{c_{p}^{\text{cancel}}}{T_{p}^{s}} \right) \sum_{p \in \hat{P}^{sC}_{hdr}} T_{p}^{s} \hat{x}_{hdp} \\
&= \left( \min_{p \in \hat{P}} \frac{c_{p}^{\text{cancel}}}{T_{p}^{s}} \right) \left( \sum_{p \in \hat{P}^{sC}_{hdr}} T_{p}^{s} \hat{x}_{hdp} - \sum_{p \in \hat{P} \setminus \hat{P}^{sC}_{hdr}} T_{p}^{s} \hat{x}_{hdp} - B_{hd}\hat{y}_{hd} \right) \\
&= \left( \min_{p \in \hat{P}} \frac{c_{p}^{\text{cancel}}}{T_{p}^{s}} \right) \left( \sum_{p \in \hat{P}} T_{p}^{s} \hat{x}_{hdp} - B_{hd}\hat{y}_{hd} \right)
\end{align*}
$$
Therefore, for any assignment $\hat{x}_{hdp} (\forall p \in \mathcal{P})$, the expression $\frac{1}{|S|} \sum_{s \in S} \left( \min_{p \in \mathcal{P}} \frac{c_{\text{cancel}}}{T_p^{sp}} \right) \left( \sum_{p \in \mathcal{P}} T_p^{s} \hat{x}_{hdp} - B_{hd} \hat{y}_{hd} \right)$ 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 (18).

**B Simple Example for BDD**

The following simple example provides an illustration of the BDD transformation. Suppose in a subproblem for an $(h, d, r, s)$ tuple there are four scheduled patients, and we need to decide whether to cancel them. The subproblem formulation is the following:

$$\begin{align*}
\text{max} & \quad 4(1 - z_{1r}^s) + (1 - z_{2r}^s) + 3(1 - z_{3r}^s) + 8(1 - z_{4r}^s) \\
\text{s.t.} & \quad 2x_1 + x_2 + 3x_3 + 3x_4 \leq 5 \\
& \quad z_{hdp}^{sp} \leq 1 \\
& \quad z_{hdp}^{sp} \in \{0, 1\}^4
\end{align*}$$

where $\hat{P}_{hdr} = \{1, 2, 3, 4\}$. The problem is reformulated via the BDD in Figure 2. The BDD contains a root node at the 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 $z^{sp}_{hdp}$, from patient 1 to patient 4. Notice that the patients can be sorted in a different way from the order in the graph, and that will produce a different BDD, thus may result in different computational performance. At each layer, the dashed and solid arcs respectively indicate selecting 0 and 1 values for the layer’s corresponding variable. The value above or beside each arc is the cost of $z^{sp}_{hdp}$ if it receives the value indicated by the arc type. When transferring the subproblem to a shortest path problem, those values are used as costs for the arcs. Each route from the node $o$ to the node $t$ represents a feasible solution for the subproblem, and each feasible solution is represented by a route from the node $o$ to the node $t$. Also, the path length provides the correct evaluation of the objective value for a corresponding feasible solution. Therefore, by finding a shortest path from the root node to the terminal node, we are able to obtain an optimal solution to the subproblem.
Figure 2: BDD for a simple subproblem with four patients

C Flow Chart for the Two-stage Decomposition

Phase one: Before branch-and-cut.

START

Add adapted FFD heuristic solution as MIP start → Generate cuts from heuristic solution → Generate constraints from subproblem relaxation → Add cuts and constraints from Phase one to initial master formulation in Phase two

Phase two: Branch-and-cut.

Apply heuristic callback

(\hat{u}, \hat{y}, \hat{x}, \hat{w}) is integral → Generate LBBBD / BDD-based Benders cuts and Classical Benders cuts

\hat{Q} < \bar{Q} or \hat{Q} < \hat{Q}^{LP}?

Yes → (\hat{u}, \hat{y}, \hat{x}, \hat{w}) is not integral → Generate Classical Benders cuts

\hat{Q} < \hat{Q}^{LP}?

No → Solve node LP relaxation

No → Stopping criterion?

Yes → STOP

Figure 3: Flow chart of the two-stage decomposition algorithm.
D Classical Benders Cuts for the Three-stage Decomposition

For the LBBD subproblem, we relax the LBBD subproblem (13). The binary variables $x_{pr}$, $y_r$, and $z_{sp}^r$ 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 optimal value of the relaxed LBBD subproblem by $\bar{Q}_{hd}^{LP}$:

$$Q_{hd}^{LP} = \min \frac{1}{|S|} \sum_{s \in S} \sum_{p \in P} \sum_{r \in R_h} c_p(x_{pr} - z_{sp}^r)$$

s.t. $\sum_{r \in R_h} x_{pr} = \hat{x}_{hdp}$ \hspace{1cm} \forall p \in P \hspace{1cm} (\gamma_p)$

$\sum_{p \in P} T^s_p z_{sp}^r \leq B_{hd} y_r$ \hspace{1cm} \forall r \in R_h, s \in S \hspace{1cm} (\beta)$

$z_{sp}^r \leq x_{pr}$ \hspace{1cm} \forall p \in P, r \in R_h, s \in S \hspace{1cm} (\iota)$

$x_{pr} \leq y_r$ \hspace{1cm} \forall p \in P \hspace{1cm} (\delta_r)$

$\sum_{r \in R_h} y_r \leq \hat{y}_{hd}$

$y_r \leq 1$ \hspace{1cm} \forall r \in R_h

$y_r, x_{pr}, z_{sp}^r \geq 0$ \hspace{1cm} \forall p \in P, r \in R_h, s \in S

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

$$Q_{hd} \geq \sum_{p \in P} \bar{\gamma}_p x_{hdp} + \bar{\beta} y_{hd} + \sum_{r \in R_h} \bar{\delta}_r$$

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 $z_{sp}^r$ as a continuous variable in subproblem (16). The optimal objective value of the relaxed subproblem is denoted by $\tilde{\theta}_{sr}^{LP}$:

$$\tilde{\theta}_{sr}^{LP} = \min \sum_{p \in P} -c_p z_{sp}^r$$

s.t. $\sum_{p \in P} T^s_p z_{sp}^r \leq B_{hd}$ \hspace{1cm} (\eta)

$z_{sp}^r \leq \hat{x}_{pr}$ \hspace{1cm} \forall p \in P \hspace{1cm} (\iota)$

$z_{sp}^r \in \{0, 1\}$ \hspace{1cm} \forall p \in P

The corresponding classical Benders cut is:

$$\theta_{sr} \geq \sum_{p \in P} \hat{\iota}_p x_{pr} + B_{hd} \hat{\eta}$$

where $\hat{\eta}$ and $\hat{\iota}_p$ are the optimal values for the corresponding dual variables.
E 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 (18) 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 (13) 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 (17); otherwise, solve the LBBD subproblem to get the optimal objective value \(\hat{Q}\). Also, solve subproblem LP relaxations and get optimal objective value \(\hat{Q}_{LP}\). 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 \(\hat{Q}\) and \(\hat{Q}_{LP}\). If \(\hat{Q} < \hat{Q}\) then add LBBD cuts; if \(\hat{Q} < \hat{Q}_{LP}\) 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 \(\hat{Q}_{LP}\), generate the classical Benders cuts and implement them if \(\hat{Q} < \hat{Q}\) 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 \(\hat{x}\) in the master solution. If \(\hat{x}\) is integral, then solve the corresponding BDD subproblems (16) and the BDD subproblem LP relaxations to get their
respective optimal objective values $\bar{\theta}$ and $\bar{\theta}^{LP}$. Generate BDD-based Benders cuts and classical Benders cuts. In the CPLEX lazy constraint callback, compare the master solution of $\bar{\theta}$ with $\bar{\theta}$ and $\bar{\theta}^{LP}$. If $\bar{\theta} < \bar{\theta}$ then add BDD-based Benders cuts; if $\bar{\theta} < \bar{\theta}^{LP}$ add classical Benders cuts. On the other hand, if some elements in the solution $\bar{\theta}$ are fractional, we only solve the subproblem LP relaxations, obtain $\bar{\theta}^{LP}$, generate the classical Benders cuts and implement them if $\bar{\theta} < \bar{\theta}^{LP}$ within the CPLEX user cut callback. The user cut management and stopping criteria are the same as in the LBBBD decomposition.

F Parameter Values for Computational Analysis

| Parameter | Value |
|-----------|-------|
| $\kappa_1$ | 50 dollars |
| $\kappa_2$ | -5 dollars |
| $\kappa_3$ | -80 dollars |
| $\kappa_4$ | -100 dollars |
| $\Gamma$ | 500 |
| $\rho_p$ | 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 |
| $\alpha_{pd}$ | Uniform distribution $[60, 120]$ days |
| $F_{hd}$ | Uniform distribution $[4000, 6000]$ |
| $G_{hd}$ | Uniform distribution $[1500, 2500]$ |