This is the author's version of an article that has been accepted for publication in the *International Journal of Production Economics*. Changes were made to this version by the publisher prior to publication. The final version of record is available at: https://doi.org/10.1016/j.ijpe.2019.07.006

Branch-and-check methods for multi-level operating room planning and scheduling

Vahid Roshanaei[+*] Kyle E. C. Booth[+†] Dionne M. Aleman[‡] David R. Urbach[§]
J. Christopher Beck[†]

September 3, 2019

Abstract

We develop the first exact decomposition approaches for a multi-level operating room planning and scheduling problem that integrates case mix planning, master surgical scheduling, and surgery sequencing in the presence of multiple surgical specialties. Our approaches consist of novel uni-level and bi-level branch-and-check algorithms that solve the problem using a hybridization of integer programming and constraint programming. We demonstrate that our approaches outperform an existing time-indexed integer programming model, yielding significant improvements on solution quality. Our methods are competitive with an existing genetic algorithm while providing provable bounds on solution quality. We conduct an investigation into the impact of time discretization on our algorithms, illustrating that our decompositions, unlike the previously proposed integer programming approach, are much less sensitive to time discretization and produce more accurate solutions as a result. Finally, we introduce and investigate benchmark instances with a more diverse case mix. Overall, we conclude that our decompositions are the most appropriate approaches for this multi-level operating room planning and scheduling problem.
Keywords: Multi-specialty operating room scheduling, logic-based Benders decomposition, branch-and-check, multi-level decomposition, hybrid IP/CP

1 Introduction

Operating rooms (ORs) play a substantial role in hospital profitability. Their effective utilization leads to cost reductions in surgical service delivery, shortening surgical patient wait times, and increasing patient admissions (Roshanaei et al., 2017b). According to a report, improving OR throughput by one additional procedure per day can generate from $4-7 million in additional annual

[*]Corresponding Author: Rotman School of Management, University of Toronto; 105 St. George, Toronto, ON M5S 3E6, Canada; vahid.roshanaei@rotman.utoronto.ca
[†]Department of Mechanical and Industrial Engineering, University of Toronto; 5 King's College Road, Toronto, ON M5S 3G8, Canada; kbooth@mie.utoronto.ca, jcb@mie.utoronto.ca
[‡]Department of Mechanical and Industrial Engineering, University of Toronto; 5 King's College Road, Toronto, ON M5S 3G8, Canada; Institute of Health Policy, Management and Evaluation, University of Toronto, 155 College Street, Suite 425, Toronto, ON M5S 3E3, Canada; Techna Institute at University Health Network; 124-100 College Street, Toronto, ON M5G 1P5, Canada; aleman@mie.utoronto.ca
[§]Division of General Surgery, Toronto General Hospital, University Health Network; 200 Elizabeth Street, Toronto, ON M5G 2C4, Canada; david.urbach@uhn.on.ca
[+]*Equally contributing authors.*

This is the author's version of an article that has been accepted for publication in the *International Journal of Production Economics*. Changes were made to this version by the publisher prior to publication. The final version of record is available at: https://doi.org/10.1016/j.ijpe.2019.07.006

revenue for an average-sized institution (HFMA, 2005). In Canada, for example, many ORs have low utilization, typically around 70% of nominal capacity due, in part, to inefficient scheduling systems, lack of downstream units for post-operative patient care, and the presence of significant uncertainty in surgical duration estimation (Wang et al., 2016). Producing high-utilization OR schedules is difficult due to the combinatorial nature of the optimization problem, which involves numerous resources (e.g., ORs, surgeons, nurses, anesthesiologists, and surgical equipment) and complex problem constraints (e.g., OR and surgeon capacities, surgery sequencing) (Denton et al., 2010). The problem is further complicated when the number of resources and patients to be scheduled increases, yielding substantial optimality gaps for solutions constructed with mathematical models and decomposition techniques (Hashemi Doulabi et al., 2016; Marques et al., 2012). In this study, we solve the multi-level OR planning and scheduling (MLORPS) problem (Marques et al., 2012) that integrates case mix planning, master surgical scheduling, and surgical case scheduling using branch-and-check (B&C) decomposition algorithms.

OR planning and scheduling includes decision-making for: i) case mix planning, at the strategic level, involving the distribution of OR times among surgical specialties (Adan et al., 2009; Vissers et al., 2005); ii) master surgical scheduling, at the tactical level, concerning the distribution of allotted OR times to surgeons within a given specialty (Banditori et al., 2013; Roshanaei et al., 2017b; Tanfani and Testi, 2010); and iii) surgical case scheduling, at the operational level (Hashemi Doulabi et al., 2016; Roshanaei et al., 2017a). Surgical case scheduling consists of advance scheduling, where the surgery date is determined, and allocation scheduling, where surgeries are allocated to ORs and their sequence is determined. At the master surgical scheduling level, the assigned OR time for each specialty is allocated to surgeons in various ways (Guerriero and Guido, 2011): block scheduling, in which an OR block, usually 8-10 hours in duration, is fully reserved for a surgeon; modified block scheduling, in which half a block is reserved for a surgeon and the remaining hours can be shared among other surgeons; and open scheduling, in which surgeons share a number of ORs and can move among them within a day (Batun et al., 2011; Hashemi Doulabi et al., 2016; Roshanaei et al., 2017b). The increased flexibility of open scheduling results in increased patient admission (Fei et al., 2009), but the scheduling problems are harder to solve (Hashemi Doulabi et al., 2016). This strategy has recently received significant attention within the OR scheduling literature (Batun et al., 2011; Fei et al., 2010; Hashemi Doulabi et al., 2016; Marques et al., 2012, 2014; Roshanaei et al., 2017b; Vijayakumar et al., 2013). In this paper, we focus exclusively on the scheduling of ORs and surgeons.

Research in OR scheduling has seen considerable activity in the last decade (Guerriero and Guido, 2011; Samudra et al., 2016; Van Riet and Demeulemeester, 2015; Zhu et al., 2018), where decision-making can be approached using heuristics (Aringhieri et al., 2015; Castro and Marques, 2015; Fei et al., 2010; Jebali et al., 2006), or exact techniques such as mathematical programming (Hashemi Doulabi et al., 2016; Marques et al., 2012; Pham and Klinkert, 2008; Roshanaei et al., 2017a; Silva et al., 2015; Vijayakumar et al., 2013) and problem-specific decompositions (Batun et al., 2011; Hashemi Doulabi et al., 2016; Riise et al., 2016; Roshanaei et al., 2017a,b). Mathematical programming is the most common approach to solving OR scheduling problems, however,

This is the author's version of an article that has been accepted for publication in the *International Journal of Production Economics*. Changes were made to this version by the publisher prior to publication. The final version of record is available at: https://doi.org/10.1016/j.ijpe.2019.07.006

recent studies have demonstrated the effectiveness of constraint programming (CP) and CP-based decomposition techniques (Hashemi Doulabi et al., 2016; Wang et al., 2015).

Case mix planning, master surgical scheduling, and surgical case scheduling have all been previously formulated using mathematical modelling, however, the simultaneous optimization of these decisions in a single mathematical model is beyond the capability of existing optimization solvers for realistically-sized problems. To handle the intractability of OR scheduling problems, a wide variety of exact techniques have been developed, including branch-and-price (Belien and Demeulemeester, 2008; Cardoen et al., 2009; Fei et al., 2008), branch-and-price-and-cut (Hashemi Doulabi et al., 2016), Lagrangian relaxation (Augusto et al., 2010; Perdomo et al., 2006), and logic-based Benders decomposition (LBBD) (Riise et al., 2016; Roshanaei et al., 2017a, 2019, 2017b). The MLORPS problem studied in this work has been previously formulated as an IP (Marques et al., 2012) and solved with CPLEX. Various heuristic algorithms have also been developed to solve the problem, including a genetic algorithm (GA) (Marques et al., 2014), and an approximate two-stage decomposition based on generalized disjunctive programming (GDP) (Castro and Marques, 2015). Previous work demonstrated that the GDP approach finds solutions 3% better than those of the IP model solved with CPLEX in one hour of runtime, and 0.40% better than the GA solutions. Bi-objective function design for MLORPS using a heuristic approach has also been studied (Marques and Captivo, 2015). Previous exact decompositions for the single-specialty variant of the problem resulted in large optimality gaps (Hashemi Doulabi et al., 2016) for problem sizes that are much smaller than those in this study. To the best of our knowledge, no exact decomposition technique has been proposed for this multi-specialty problem.

The MLORPS problem we solve represents one of the largest integrated OR planning and scheduling problems under open scheduling, with instances exceeding 1000 patients. Recognizing the shortcomings of the IP model for problems of this size, we propose the use of exact branch-and-check (B&C) decomposition methods. Logic-based Benders decomposition (LBBD) (Hooker, 1994; Hooker and Ottosson, 2003) and B&C (Thorsteinsson, 2001) are decomposition algorithms that have been applied successfully to a variety of scheduling problems, including single and multi-stage parallel machine scheduling (Harjunkoski and Grossmann, 2002; Tran et al., 2016), multi-factory planning and scheduling (Hooker, 2007), home hospice care scheduling (Heching et al., 2019), job availability intervals (Gedik et al., 2016), vehicle routing (Booth et al., 2016), OR scheduling (Riise et al., 2016; Roshanaei et al., 2017a, 2019, 2017b), integrated process planning and scheduling (Barzanji et al., 2019), and multi-period interdiction networks (Enayaty-Ahangar et al., 2018). The B&C algorithm, specifically, is closely related to both classical Benders decomposition and LBBD. Classical Benders decomposition partitions a mixed-integer program into a master problem (MP), involving only integer variables, and one or more subproblems (SPs) involving only continuous variables. LBBD generalizes classical Benders decomposition by allowing the SPs to take on any form. While both classical and logic-based Benders approaches solve the MP to optimality and then solve the SPs given the fixed MP-optimal solution, B&C solves the SPs during the MP search for each incumbent solution and can thus guarantee a bound for each incumbent solution if verified with SPs. A variant of B&C (OPT15) (Beck, 2010) only solves SPs when the optimality gap of

This is the author's version of an article that has been accepted for publication in the *International Journal of Production Economics*. Changes were made to this version by the publisher prior to publication. The final version of record is available at: https://doi.org/10.1016/j.ijpe.2019.07.006

MP integer solutions is equal to or less than 15%, resulting in significant computational savings.

It has been previously shown that operating room scheduling problems are amenable to allocation-sequencing (Castro and Marques, 2015; Jebali et al., 2006; Riise et al., 2016; Roshanaei et al., 2017b); allocation-packing (Roshanaei et al., 2017a); and/or allocation-packing-balancing (Roshanaei et al., 2019) decompositions. We develop uni-level and bi-level B&C approaches to solve the MLORPS problem as initially formulated by Marques et al. (2012). The uni-level B&C approach decomposes the problem into an MP, consisting of case mix selection, OR-to-specialty allocation, surgery-to-OR allocation, and a set of SPs, where surgeries are sequenced. In the bi-level B&C approach, the MP is relaxed to only include case mix selection and relaxed OR-to-specialty allocation, which we call the relaxed MP (RMP). The primary SPs allocate surgeries to ORs and ensure that surgeries can be packed within the RMP-determined number of ORs. The secondary SPs sequence surgeries in ORs allocated to each surgical specialty in each day. We model the MP, RMP, and primary SPs using IP and the secondary SPs using CP. We compare the performance of our decomposition methods to those of the IP model solved via CPLEX (Marques et al., 2012) and a GA (Marques et al., 2014) using the same dataset over two objective functions: i) maximizing the *scheduled surgical time* and ii) maximizing *the number of scheduled surgeries*. A direct comparison to the approximate GDP decomposition (Castro and Marques, 2015) is not possible as a different dataset was used that is not accessible to us.

The contributions of this paper are as follows:

- We develop the first exact decomposition algorithms for the multi-level operating room planning and scheduling (MLORPS) problem, specifying associated master problems, subproblems, and logic-based Benders cuts. We prove the validity of our Benders cuts and the global convergence of our approaches.
- We model and solve surgery sequencing subproblems with constraint programming (CP), leveraging its ability to rapidly determine feasible solutions to these hard scheduling problems.
- We demonstrate that our uni-level decompositions provide, on average, 10 times smaller optimality gaps than the integer programming (IP) model solved with CPLEX (Marques et al., 2012) within a 30 minute runtime. Similarly, our decomposition methods are able to find feasible solutions competitive with a genetic algorithm (GA) (Marques et al., 2014) in less runtime while providing provable bounds on solution quality.
- We show that, in addition to outperforming the existing IP model on time-discretized problems, our decomposition methods are capable of solving MLORPS effectively with non-discretized surgical/OR times. Furthermore, we show that arbitrary discretization of surgical times and OR availability times, as implemented in previous approaches in the literature, leads to an inaccurate estimation of OR utilization.
- We validate the performance of our approaches on both existing benchmarks and newly introduced benchmark data with a more diverse case mix, demonstrating the performance achieved is similar to that of the existing benchmark.

The remainder of the paper is structured as follows. In Section 2, we formalize the problem and provide the mathematical model. In Section 3, we present our uni-level and bi-level decomposition approaches. In Sections 4, we provide a description of the data used for our experiments. In Section 5, we present various experimental analyses, including a comparison to previous methods used to solve MLORPS. In Section 6, we discuss the results and Section 7 concludes the paper. Proofs of the Benders cuts are provided in the Appendix.

2 Problem Definition

The MLORPS problem, as presented in Marques et al. (2012), is defined as follows: Given a set of patients, a set of surgical specialties, a set of homogeneous ORs, and a set of surgeons, develop a schedule for a fixed planning horizon that maximizes OR utilization (or, alternatively, the number of scheduled surgeries) while adhering to OR availability, surgeon availability, and patient priorities. In the original problem definition, although ORs are homogeneous, they cannot be shared among specialties and each OR must be allocated to a single specialty on each day. Each surgery has a surgical time, where induction time, surgery time, and wake-up time are aggregated into a single value. There is also a mandatory cleaning time that takes place after each surgery. Surgeons are assigned to surgeries *a priori* and a surgeon may participate in surgeries from multiple specialties. In Marques et al. (2012), time is discretized into time slots of 15 minutes. In this work, in addition to studying the discretized variant, we investigate solving the non-discretized problem. Time slots during which surgeons are unavailable are also considered. Deadlines for surgeries are determined by patient priority classification: i) deferred urgency (surgery must be performed in ≤ 72 hours), ii) high priority (surgery must be completed within a fortnight), iii) priority (surgery must be completed within two months), and iv) normal priority (within a year).

An IP formulation, based on the one presented in Marques et al. (2012), is used to define and solve the problem. With some adjustments to the original presentation, intended for clarity throughout this work, the notation for all parameters and variables in this model is summarized in Table 1. The problem is modeled following two assumptions often made in the literature: i) surgical times are deterministic and known *a priori* (Fei et al., 2009; Hashemi Doulabi et al., 2016; Jebali et al., 2006; Riise et al., 2016; Roshanaei et al., 2017a,b), and ii) ORs are homogeneous (Castro and Marques, 2015; Marques et al., 2014; Roshanaei et al., 2017a). While this work both presents, and compares against, deterministic algorithms for MLORPS, there is recent work on the incorporation of surgical duration stochasticity into OR scheduling problems (Guerriero and Guido, 2011; Rath et al., 2017; Samudra et al., 2016; Shylo et al., 2013; Van Riet and Demeulemeester, 2015; Zhu et al., 2018). The IP model is based on an open scheduling strategy, allowing surgeons to change ORs within a day to perform their assigned surgeries and assumes that surgeons are always available during OR availability times. Surgeons are not required to stay in the OR during cleaning time. This model integrates case mix planning, OR-to-specialty allocation, and patient-to-OR allocation with patient sequencing.

This is the author's version of an article that has been accepted for publication in the *International Journal of Production Economics*. Changes were made to this version by the publisher prior to publication. The final version of record is available at: https://doi.org/10.1016/j.ijpe.2019.07.006

Table 1: IP model notation

Sets:	
$\mathcal{J}$	Set of surgical specialties, $j \in \mathcal{J}$
$\mathcal{S}$	Set of surgeons, $s \in \mathcal{S}$
$\mathcal{P}$	Set of patients, $p \in \mathcal{P}$
$\mathcal{P}_1$	Set of mandatory patients that must be operated on in the first day of the week
$\mathcal{P}_2$	Set of mandatory patients that must be operated on sometime during the week
$\mathcal{S}_j$	Set of surgeons associated with specialty j
$\mathcal{P}_j$	Set of patients associated with specialty j
s_p	Surgeon assigned to patient p, $s_p \in \mathcal{S}$
$\mathcal{D}$	Set of days in the planning horizon, $d \in \mathcal{D}$
$\mathcal{R}$	Set of ORs in the surgical suite of each hospital, $r \in \mathcal{R}$
$\mathcal{L}$	Set of time slots representing the available time of ORs, $t \in \mathcal{L}$
$\mathcal{L}_p$	A subset of time slots for which surgery p can start without going into overtime
Parameters:	
T_p	Total surgical duration of patient p
E	Fixed cleaning time after each surgery
A_{sd}	Maximum availability time of surgeon s on day d
A_s	Maximum availability time of surgeon s in each week
Variables:	
x_{pdrt}	1 if the surgery of patient p starts in slot t of OR r on day d, 0 otherwise
y_{jdr}	1 if specialty j is allocated to OR r on day d, 0 otherwise

2.1 Constraints

Following Marques et al. (2012), the constraints required to formulate the IP-MLORPS are detailed by Constraints (1)-(11). Constraint (1) ensures that the set of deferred urgency patients, $\mathcal{P}_1$, are scheduled on the first day of the planning horizon, while Constraint (2) ensures that the set of high priority patients, $\mathcal{P}_2$, are scheduled in the current planning horizon. Constraint (3) models the set of priority and normal priority patients, $\mathcal{P} \setminus \{\mathcal{P}_1 \cup \mathcal{P}_2\}$, as optional in the current planning horizon. Constraint (4) ensures that no two surgeries interfere temporally in an OR. Constraint (5) ensures that each OR is only allocated to a single specialty on each day. Constraint (6) ensures that the maximum availability of each OR is not exceeded for each day. Constraint (7) ensures that surgeries involving the same surgeon do not temporally interfere. Constraints (8) and (9) ensure the daily and weekly availability times of each surgeon are respected. Finally, Constraints (10) and (11) identify the binary domain of the decision variables.

This is the author's version of an article that has been accepted for publication in the *International Journal of Production Economics*. Changes were made to this version by the publisher prior to publication. The final version of record is available at: https://doi.org/10.1016/j.ijpe.2019.07.006

$$
\begin{aligned}
\text{maximize}\quad & \text{Objective functions in Section 2.2} && && \text{(IP-MLORPS)}\\
\text{subject to}\quad & \sum_{r\in\mathcal{R}}\sum_{t\in\mathcal{L}_p} x_{p1rt} = 1 && \forall p \in \mathcal{P}_1 && (1)\\
& \sum_{d\in\mathcal{D}}\sum_{r\in\mathcal{R}}\sum_{t\in\mathcal{L}_p} x_{pdrt} = 1 && \forall p \in \mathcal{P}_2 && (2)\\
& \sum_{d\in\mathcal{D}}\sum_{r\in\mathcal{R}}\sum_{t\in\mathcal{L}_p} x_{pdrt} \le 1 && \forall p \in \mathcal{P} \setminus \{\mathcal{P}_1 \cup \mathcal{P}_2\} && (3)\\
& \sum_{p\in\mathcal{P}}\sum_{t'\in\mathcal{L}_p : t'\in[t-T_p+1-E,t]} x_{pdrt'} \le 1 && \forall d\in\mathcal{D};\ r\in\mathcal{R};\ t\in\mathcal{L} && (4)\\
& \sum_{j\in\mathcal{J}} y_{jdr} \le 1 && \forall d\in\mathcal{D};\ r\in\mathcal{R} && (5)\\
& \sum_{p\in\mathcal{P}_j}\sum_{t\in\mathcal{L}_p} x_{pdrt} \le |\mathcal{L}| y_{jdr} && \forall j\in\mathcal{J};\ d\in\mathcal{D};\ r\in\mathcal{R} && (6)\\
& \sum_{p\in\mathcal{P}:s_p=s}\sum_{r\in\mathcal{R}}\sum_{t'\in\mathcal{L}_p:t'\in[t-T_p+1,t]} x_{pdrt'} \le 1 && \forall s\in\mathcal{S};\ d\in\mathcal{D};\ t\in\mathcal{L} && (7)\\
& \sum_{p\in\mathcal{P}:s_p=s}\sum_{r\in\mathcal{R}}\sum_{t\in\mathcal{L}_p} T_p x_{pdrt} \le A_{sd} && \forall s\in\mathcal{S};\ d\in\mathcal{D} && (8)\\
& \sum_{p\in\mathcal{P}:s_p=s}\sum_{d\in\mathcal{D}}\sum_{r\in\mathcal{R}}\sum_{t\in\mathcal{L}_p} T_p x_{pdrt} \le A_s && \forall s\in\mathcal{S} && (9)\\
& x_{pdrt} \in \{0,1\} && \forall p\in\mathcal{P};\ d\in\mathcal{D};\ r\in\mathcal{R};\ t\in\mathcal{L}_p && (10)\\
& y_{jdr} \in \{0,1\} && \forall j\in\mathcal{J}; d\in\mathcal{D}; r\in\mathcal{R} && (11)
\end{aligned}
$$

2.2 Objective functions

With the problem constraints defined in Section 2.1, two objective functions have been considered for the IP-MLORPS model in the literature, aimed at maximizing: i) scheduled surgical time (Marques et al., 2012), and ii) the number of scheduled surgeries (Marques et al., 2014). The first objective is formulated as follows:

$$\sum_{p\in\mathcal{P}}\sum_{d\in\mathcal{D}}\sum_{r\in\mathcal{R}}\sum_{t\in\mathcal{L}_p} T_p x_{pdrt} \tag{12}$$

effectively summing all of the surgical time scheduled across the patients, days, and ORs in the planning horizon. By removing the coefficient T_p, we can formulate the second objective, namely the number of scheduled surgeries:

$$\sum_{p\in\mathcal{P}}\sum_{d\in\mathcal{D}}\sum_{r\in\mathcal{R}}\sum_{t\in\mathcal{L}_p} x_{pdrt} \tag{13}$$

We note that while various objective functions can be designed and tested for the constraints in the IP-MLORPS model, we study these two objective functions to be consistent with the literature

This is the author's version of an article that has been accepted for publication in the *International Journal of Production Economics.* Changes were made to this version by the publisher prior to publication. The final version of record is available at: https://doi.org/10.1016/j.ijpe.2019.07.006

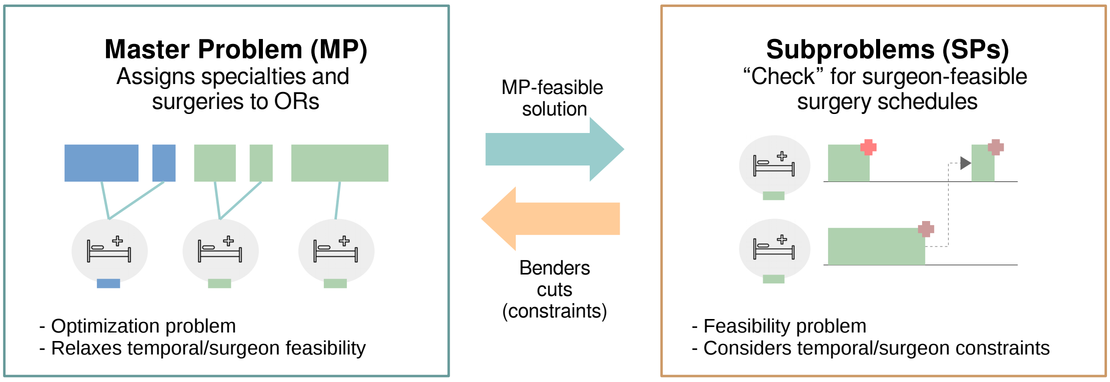

Figure 1: High-level branch-and-check (B&C) components for MLORPS, including the master problem (MP), sub problems (SPs), and Benders cuts.

and for comparison purposes. To motivate and present our decomposition methods, we use only the scheduled surgical time maximization objective function.

3 Proposed Decomposition Approaches

In this section, we detail the general algorithmic procedure for our decomposition methods and present our uni-level and bi-level approaches to the MLORPS problem.

3.1 Background

MLORPS is an $\mathcal{NP}$-hard problem (Marques, 2010) and the IP-MLORPS formulation has $\mathcal{O}(|\mathcal{P}| \times |\mathcal{D}| \times |\mathcal{R}| \times |\mathcal{L}|)$ binary variables and $\mathcal{O}(|\mathcal{S}| \times |\mathcal{D}| \times |\mathcal{L}|)$ constraints (assuming $|\mathcal{S}| >> |\mathcal{J}|$ and $|\mathcal{L}| >> |\mathcal{R}|$). The time-indexed nature of the formulation, with a constraint posted for each $t \in \mathcal{L}$, poses challenges for modern solvers, as we demonstrate in later sections of the paper: memory usage grows dramatically and performance deteriorates quickly as the size of the problem increases. To mitigate this, we investigate the use of row-generation techniques, specifically branch-and-check (B&C) (Thorsteinsson, 2001), as an alternative to the existing time-indexed formulation.

In B&C, as illustrated in Figure 1, the global problem (i.e., IP-MLORPS) is relaxed by either omitting constraints and/or redefining decision variables to form a master problem (MP) that is expressed as a mathematical program. In the context of our MLORPS problem, the MP relaxes surgery sequencing constraints and simply assigns ORs to specialties and surgeries to ORs subject to OR capacities and surgeon availability. The MP is solved using a branch-and-bound solver and, whenever a feasible solution is found, it is verified ("checked") against the subproblems (SPs). The purpose of the SPs is to enforce the problem components that were relaxed in the formation of the MP. For our problem, the SPs verify whether the surgeries assigned to a set of ORs on a given day can be feasibly sequenced given surgeon and OR availability. If the SP finds that the MP solution violates its constraints, it returns a cut to the MP indicating that the solution is not globally feasible. If the MP-feasible solution is also feasible with respect to all SPs, then it is globally valid

This is the author's version of an article that has been accepted for publication in the *International Journal of Production Economics*. Changes were made to this version by the publisher prior to publication. The final version of record is available at: https://doi.org/10.1016/j.ijpe.2019.07.006

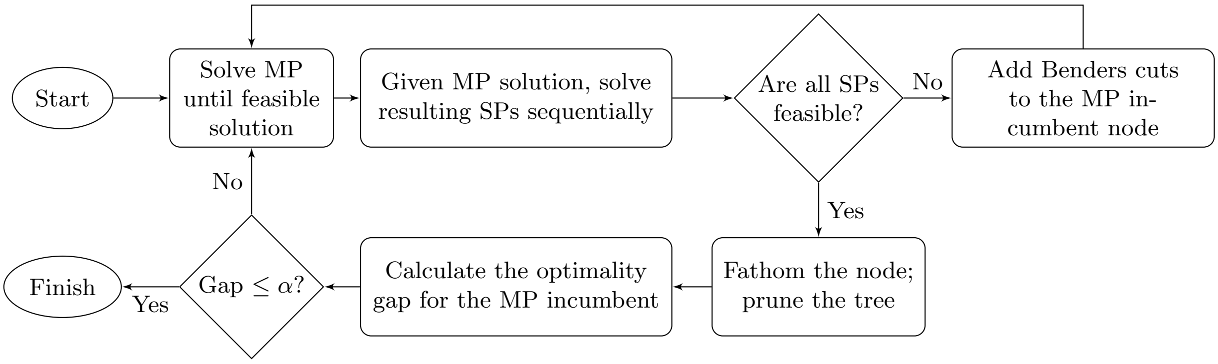

Figure 2: Uni-level branch-and-check (B&C) algorithm detail. Parameter α represents the acceptable optimality gap of the decision maker.

and the MP search continues until terminated by the runtime limit, or optimality is proven. As formal definitions of the B&C procedure can be found in previous works (Thorsteinsson, 2001; Tran et al., 2016), we provide a conceptual understanding of the algorithm and insight for how it is used to solve the MLORPS problem.

3.2 Uni-level Branch-and-Check

In the uni-level B&C, the MP optimizes the assignment of decision variables, x_{pdr}, which simultaneously determine the case mix and case-to-OR allocation. Variables y_{jdr} determine which ORs should be allocated to each specialty on each day. The sequencing SPs (SSPs) look to feasibly sequence the surgeries allocated to each day, subject to surgeon and OR availability. The uni-level B&C approach converges to optimality when the MP lower bound is equal to its upper bound, where the upper bound is given by the best MP-feasible solution that has also passed all of the SP checks.

3.2.1 Master Problem

The uni-level MP removes index t from x_{pdrt}, as well as the constraints that enforce feasible sequencing among the time slots of each OR and surgeon (Constraints (4) and (7)). Unlike the IP-MLORPS model, wherein surgical and cleaning times of each surgery are expressed as a pre-specified number of discrete time slots, the uni-level MP can use continuous surgical and cleaning times, leading to a more accurate representation. The MP model is detailed as follows, with objective function and Constraints (14)-(24).

$$\text{maximize} \quad \sum_{p\in\mathcal{P}}\sum_{d\in\mathcal{D}}\sum_{r\in\mathcal{R}} T_p x_{pdr} \tag{MP}$$

$$\text{subject to} \quad \sum_{r\in\mathcal{R}} x_{p1r} = 1 \qquad \forall p \in \mathcal{P}_1 \tag{14}$$

$$\sum_{d\in\mathcal{D}}\sum_{r\in\mathcal{R}} x_{pdr} = 1 \qquad \forall p \in \mathcal{P}_2 \tag{15}$$

This is the author's version of an article that has been accepted for publication in the *International Journal of Production Economics*. Changes were made to this version by the publisher prior to publication. The final version of record is available at: https://doi.org/10.1016/j.ijpe.2019.07.006

$$\sum_{d\in\mathcal{D}}\sum_{r\in\mathcal{R}} x_{pdr} \le 1 \qquad \forall p \in \mathcal{P} \setminus \{\mathcal{P}_1 \cup \mathcal{P}_2\} \tag{16}$$

$$\sum_{j\in\mathcal{J}} y_{jdr} \le 1 \qquad \forall d \in \mathcal{D};\ r \in \mathcal{R} \tag{17}$$

$$\sum_{p\in\mathcal{P}_j} (T_p + E)x_{pdr} \le (B + E)y_{jdr} \qquad \forall j \in \mathcal{J};\ d \in \mathcal{D};\ r \in \mathcal{R} \tag{18}$$

$$\sum_{p\in\mathcal{P}:s_p=s}\sum_{r\in\mathcal{R}} T_p x_{pdr} \le A_{sd} \qquad \forall s \in \mathcal{S};\ d \in \mathcal{D} \tag{19}$$

$$\sum_{p\in\mathcal{P}:s_p=s}\sum_{d\in\mathcal{D}}\sum_{r\in\mathcal{R}} T_p x_{pdr} \le A_s \qquad \forall s \in \mathcal{S} \tag{20}$$

$$\sum_{p\in\mathcal{P}}\sum_{r\in\mathcal{R}} x_{pdr} \ge \sum_{p\in\mathcal{P}}\sum_{r\in\mathcal{R}} x_{pd'r} \qquad \forall d = 2, \ldots, |\mathcal{D}| - 1; d < d' \le |\mathcal{D}| \tag{21}$$

$$\sum_{d\in\mathcal{D}}\sum_{r\in\mathcal{R}} x_{pdr} \ge \sum_{d\in\mathcal{D}}\sum_{r\in\mathcal{R}} x_{p'dr} \qquad \forall p \in (\mathcal{P} : s_p = s'_p, T_p = T'_p); p < p' \le |\mathcal{P}| \tag{22}$$

$$x_{pdr} \in \{0, 1\} \qquad \forall p \in \mathcal{P};\ d \in \mathcal{D};\ r \in \mathcal{R} \tag{23}$$

$$y_{jdr} \in \{0, 1\} \qquad \forall j \in \mathcal{J};\ d \in \mathcal{D};\ r \in \mathcal{R} \tag{24}$$

While most of the constraints remain similar to the IP-MLORPS model, Constraint (18) ensures that the time required for each surgery includes both the surgical time (T_p) and cleaning time (E). Due to symmetric OR (720 minutes: 690 minutes for surgery allocation and 30 minutes for the cleaning time of the last surgery) and surgeon (360 minutes) availability in the model, the MP may interchange the set of patients among days leading to different solutions with the same objective function value. Constraint (21) breaks this symmetry by ensuring the number of scheduled surgeries per day is ordered. Deferred priority patients, $\mathcal{P}_1$, are not considered in the symmetry-breaking constraint because they must be operated on on the first day of the week, which removes symmetry. Similarly, Constraint (22) breaks the symmetry among ORs on each day. Because two or more surgeries with equivalent duration assigned to the same surgeon are effectively the same, this constraint pre-orders the use of these surgeries to remove symmetrical solutions with the same objective function.

3.2.2 Subproblem

Notation for the uni-level SP is shown in Table 2. Each MP incumbent solution i provides the following output to specialty j on day d: a set of (i) patients, $\hat{\mathcal{P}}_{jd}^{(i)}$, (ii) surgeons, $\hat{\mathcal{S}}_{jd}^{(i)}$, and (iii) ORs, $\hat{\mathcal{R}}_{jd}^{(i)}$. Specialties that share surgeons in the incumbent are grouped together recursively and an SSP is constructed for the specialty group (e.g., if specialties A and B share a surgeon, and specialties B and C share a surgeon, the generated specialty group is {A,B,C}). Specialties whose surgeons exclusively work for it form an SSP on their own. The set $\tilde{\mathcal{J}}_d^{(i)}$ denotes the specialty groups for incumbent i on day d. An SSP, formulated as a CP, is then solved for each specialty group, $\hat{\mathcal{J}}_d^{(i)} \in \tilde{\mathcal{J}}_d^{(i)}$, that has been assigned more than one OR in the incumbent solution.

This is the author's version of an article that has been accepted for publication in the *International Journal of Production Economics*. Changes were made to this version by the publisher prior to publication. The final version of record is available at: https://doi.org/10.1016/j.ijpe.2019.07.006

Table 2: Uni-level SP notation

Sets:	
$\hat{\mathcal{P}}^{(i)}_{jd}$	All patients p from specialty j scheduled on day d by the MP for incumbent i
$\hat{\mathcal{R}}^{(i)}_{jd}$	All ORs r assigned to specialty j on day d by the MP for incumbent i
$\hat{\mathcal{J}}^{(i)}_{d}$	Specialty group (formed for incumbent i on day d) such that each specialty, $j \in \hat{\mathcal{J}}^{(i)}_{d}$, shares at least one surgeon with another specialty in the group
$\tilde{\mathcal{J}}^{(i)}_{d}$	Set of specialty groups $\hat{\mathcal{J}}^{(i)}_{d}$ on day d for incumbent i
$\hat{\mathcal{S}}^{(i)}_{\hat{\mathcal{J}}d}$	All surgeons s from specialty group $\hat{\mathcal{J}}^{(i)}_{d}$ operating on day d by the MP for incumbent i
$\hat{\mathcal{P}}^{(i)}_{\hat{\mathcal{J}}sd}$	All patients from $\bigcup_{j \in \hat{\mathcal{J}}^{(i)}_{d}} \hat{\mathcal{P}}^{(i)}_{jd}$ treated by surgeon s on day d by the MP for incumbent i
Parameters:	
B	The maximum availability time for each OR block
E	Fixed cleaning time after each surgery
Variables:	
a_p	Interval variable for patient p's case, defined by a time interval and duration, T_p
$\bar{a}_{pr}$	Optional interval variable for patient p's case in room r, and includes cleaning time, defined by an interval and duration $T_p + E$

$$\underset{a,\bar{a}}{\text{minimize}} \quad \mathbf{0} \qquad \text{(SSP)}$$

$$\text{subject to} \quad \sum_{r \in \hat{\mathcal{R}}^{(i)}_{jd}} \text{PresenceOf}(\bar{a}_{pr}) = 1 \qquad \forall j \in \hat{\mathcal{J}}^{(i)}_{d}; p \in \hat{\mathcal{P}}^{(i)}_{jd} \qquad (25)$$

$$\text{NoOverlap}\left(\left\{\bar{a}_{pr} : p \in \hat{\mathcal{P}}^{(i)}_{jd}\right\}\right) \qquad \forall j \in \hat{\mathcal{J}}^{(i)}_{d}; r \in \hat{\mathcal{R}}^{(i)}_{jd} \qquad (26)$$

$$\text{NoOverlap}\left(\left\{a_{p} : p \in \hat{\mathcal{P}}^{(i)}_{\hat{\mathcal{J}}sd}\right\}\right) \qquad \forall s \in \hat{\mathcal{S}}^{(i)}_{\hat{\mathcal{J}}d} \qquad (27)$$

$$\text{StartAtStart}(a_p, \bar{a}_{pr}, 0) \qquad \forall j \in \hat{\mathcal{J}}^{(i)}_{d}; p \in \hat{\mathcal{P}}^{(i)}_{jd}; r \in \hat{\mathcal{R}}^{(i)}_{jd} \qquad (28)$$

$$\text{PresenceOf}(\bar{a}_{pr}) \in \{0, 1\} \qquad \forall j \in \hat{\mathcal{J}}^{(i)}_{d}; p \in \hat{\mathcal{P}}^{(i)}_{jd}; r \in \hat{\mathcal{R}}^{(i)}_{jd} \qquad (29)$$

$$0 \leq \text{Start}(a_p) \leq B - T_p \qquad \forall j \in \hat{\mathcal{J}}^{(i)}_{d}; p \in \hat{\mathcal{P}}^{(i)}_{jd} \qquad (30)$$

The objective function represents the feasibility objective of the problem. The SSP makes use of both interval variables and optional interval variables, rich variable types available within CP. An interval variable (Laborie, 2009) is a decision variable whose possible values are convex intervals: $\{\perp\} \cup \{(f, g) \mid f, g \in \mathbb{Z}, f \leq g\}$, where f and g are the start and end values of the interval and $\perp$ is a special value that indicates the variable is not present in the solution. Interval variable $a_p, \forall p \in \hat{\mathcal{P}}^{(i)}_{jd}$, is also defined by a processing time, T_p. Each patient/OR pair (p, r) has an optional interval variable with processing time $T_p + E$. Constraint (25) ensures that each surgery is assigned to one OR by ensuring that only one $\bar{a}_{pr}$ variable is present, i.e., equal to 1. Constraint (26) uses the NoOverlap global constraint (Baptiste et al., 2001) to ensure that the surgeries assigned to a particular room do not overlap, while accounting for cleaning time. Constraint (27) uses similar reasoning to ensure that interval variables associated with each surgeon do not interfere temporally, excluding cleaning time. Constraint (28) uses the constraint StartAtStart to ensure the two types of interval variables are synchronized with respect to the start time. Constraint (30) defines the domain for the start time of each interval variable. To strengthen constraint propagation within the CP solver, we fix the assignment of the largest surgery in a particular SSP to the first OR and break symmetries among identical surgeries. Such decisions do not eliminate the SSP optimum, but reduce the assignment and start time domains on the remaining interval variables.

If an SSP is feasible, the current assignment of patients to ORs and allocation of ORs to specialties on day d is globally valid. If, however, the SSP model is infeasible, we must then introduce a Benders cut to the MP. We let $\bar{\mathcal{J}}_d^{(i)} \subseteq \tilde{\mathcal{J}}_d^{(i)}$ denote the set of specialty groups whose SSPs are infeasible.

3.2.3 Cuts

We add a Benders cut for each specialty group whose MP allocation for incumbent i results in an infeasible SSP. The Benders cut associated with each infeasible SSP is as follows:

$$\left(\left| \bigcup_{j \in \hat{\mathcal{J}}_d^{(i)}} \hat{\mathcal{P}}_{jd}^{(i)} \right| - \sum_{j \in \hat{\mathcal{J}}_d^{(i)}} \sum_{p \in \hat{\mathcal{P}}_{jd}^{(i)}} \sum_{r \in \hat{\mathcal{R}}_{jd}^{(i)}} x_{pdr} \right) + \left(\sum_{j \in \hat{\mathcal{J}}_d^{(i)}} \sum_{r \in \mathcal{R}} y_{jdr} - \left| \bigcup_{j \in \hat{\mathcal{J}}_d^{(i)}} \hat{\mathcal{R}}_{jd}^{(i)} \right| \right) \geq 1 \qquad \forall \hat{\mathcal{J}}_d^{(i)} \in \bar{\mathcal{J}}_d^{(i)} \tag{31}$$

Inequality (31) is a valid Benders cut (Theorem 1) that breaks the SSP infeasibility by ensuring at least one patient is removed from $\bigcup_{j \in \hat{\mathcal{J}}_d^{(i)}} \hat{\mathcal{P}}_{jd}^{(i)}$ (MP patient allocation) and/or at least one more OR is added to $\bigcup_{j \in \hat{\mathcal{J}}_d^{(i)}} \hat{\mathcal{R}}_{jd}^{(i)}$ (MP OR allocation). If $\left| \bigcup_{j \in \hat{\mathcal{J}}_d^{(i)}} \hat{\mathcal{R}}_{jd}^{(i)} \right| = |\mathcal{R}|$ in the infeasible SSP, the MP can only break the SSP infeasibility by reducing the number of patients.

Each valid Benders cut must satisfy two properties (Chu and Xia, 2004): i) It must eliminate the current MP solution, and ii) It must not remove any other globally feasible integer solutions. We use an approach similar to (Roshanaei et al., 2017a,b) to prove the validity of this cut.

Theorem 1. *Inequality (31) is a valid Benders cut. The proof is provided in Appendix (1).*

3.3 Bi-level Branch-and-Check

The structure of the MP, due to the assumption of identical ORs, makes another level of decomposition possible (Figure 3). The decision whether to select a patient for operation in the current planning horizon (case mix selection) does not necessitate specific patient-to-OR allocation and can be done by only determining the date of surgery for each patient instead. Since patient surgical loads ($T_p + E$) for each OR and patient-to-specialty allocation are known a priori, we can later determine the minimum number of ORs required to accommodate $\hat{\mathcal{P}}_{jd}^{(i)}$ as an independent bin (OR)

This is the author's version of an article that has been accepted for publication in the *International Journal of Production Economics*. Changes were made to this version by the publisher prior to publication. The final version of record is available at: https://doi.org/10.1016/j.ijpe.2019.07.006

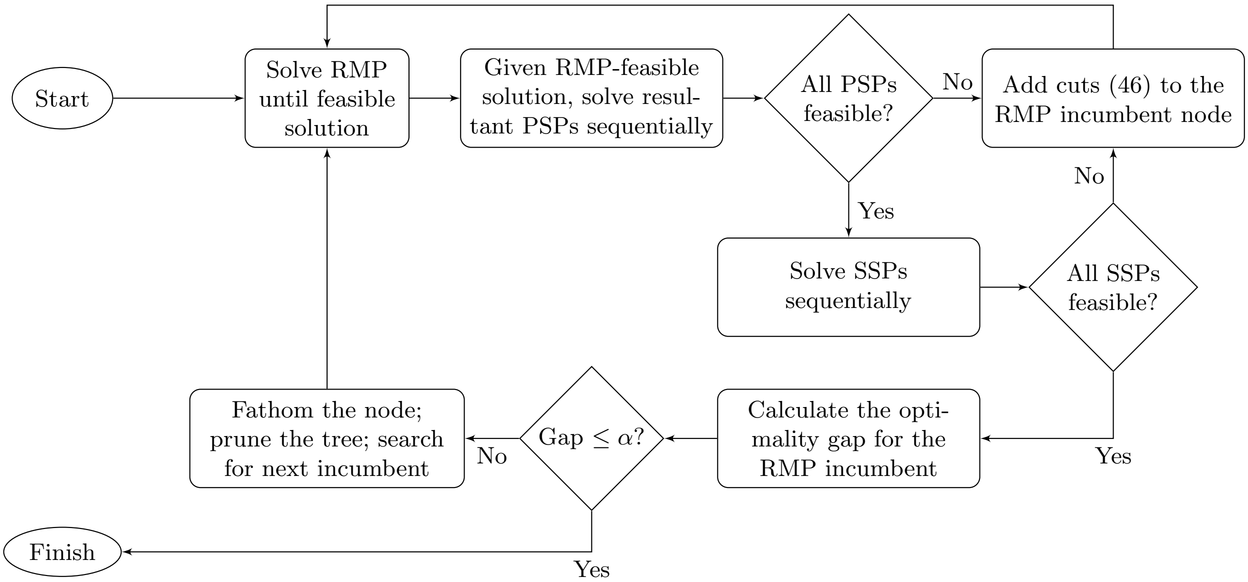

Figure 3: Bi-level branch-and-check (B&C) algorithm detail. Parameter α represents the acceptable optimality gap of the decision maker.

packing optimization problem (Fazel-Zarandi and Beck, 2012; Roshanaei et al., 2017a). Therefore, we can decompose the MP into a relaxed MP (RMP), where the case mix of patients is determined, and a relaxed packing subproblem (PSP) that determines a lower bound on the number of ORs for each specialty on each day. The feasible assignment of allocated patients to ORs and time slots within each OR is done in the PSP and SSP, respectively. If an RMP incumbent i passes all PSP checks, the incumbent is given to the SSP to ensure feasible surgery sequencing. The RMP incumbent solution i is globally valid if it is feasible with respect to all PSPs and SSPs. The bi-level B&C approach converges to optimality if the gap of the RMP incumbent i is $\leq \alpha$ (optimality tolerance); otherwise, the search continues for the next incumbent with a lower optimality gap.

The rationale behind the additional level of decomposition is to reduce the number of indices in the assignment binary variables from three indices to two ($x_{pdr} \rightarrow x_{pd}$), reducing the difficulty of solving the RMP. The bi-level B&C resembles the real-world OR scheduling decision making process more closely than the uni-level B&C in that, in the real world, the patient is first selected for surgery (case mix planning) and the date of surgery is later determined (advance scheduling). Then, surgery-to-OR allocation is determined (first stage of allocation scheduling), and finally, surgeries are assigned to a starting time slot within ORs (second stage of allocation scheduling). Since patient-to-surgeon assignments are known a priori, optimizing x_{pd} and later x_{pr} determines how much a surgeon should work in that day (master surgical scheduling). The drawback of removing index r is that PSPs are frequently solved for each RMP incumbent solution, which represents a trade-off between the computational effort the RMP expends branching on index r compared to the time required to solve PSPs.

This is the author's version of an article that has been accepted for publication in the *International Journal of Production Economics*. Changes were made to this version by the publisher prior to publication. The final version of record is available at: https://doi.org/10.1016/j.ijpe.2019.07.006

3.3.1 Relaxed Master Problem

The RMP removes index r from the assignment variable x_{pdr} in MP and uses a two-indexed assignment variable x_{pd} to only assign patients to days. The RMP also removes index r from the OR allocation decision variable y_{jdr} and uses a two-indexed decision variable y_{jd}. Determination of patient-to-day assignments for each specialty on each day allows the RMP to ascertain the number of required ORs for that specialty by dividing the total surgical load $(T_p + E)$ by the OR availability time on each day $(B + E)$ (Fazel-Zarandi and Beck, 2012). The RMP is detailed by objective function and Constraints (32) through (42).

$$
\begin{aligned}
\underset{\mathbf{x},\mathbf{y}}{\text{maximize}} \quad & \sum_{p\in\mathcal{P}}\sum_{d\in\mathcal{D}} T_p x_{pd} && && \text{(RMP)}\\
\text{subject to} \quad & \sum_{d\in\mathcal{D}} x_{pd} = 1 && \forall p \in \mathcal{P}_2 && (32)\\
& \sum_{d\in\mathcal{D}} x_{pd} \leq 1 && \forall p \in \mathcal{P} \setminus \{\mathcal{P}_1 \cup \mathcal{P}_2\} && (33)\\
& \frac{\sum_{p\in\mathcal{P}_j}(T_p + E)x_{pd}}{B + E} \leq y_{jd} \leq |\mathcal{R}| && \forall j \in \mathcal{J};\ d \in \mathcal{D} && (34)\\
& \sum_{j\in\mathcal{J}} y_{jd} \leq |\mathcal{R}| && \forall d \in \mathcal{D} && (35)\\
& \sum_{p\in\mathcal{P}_s} T_p x_{pd} \leq A_{sd} && \forall s \in \mathcal{S};\ d \in \mathcal{D} && (36)\\
& \sum_{p\in\mathcal{P}:s_p=s}\sum_{d\in\mathcal{D}} T_p x_{pd} \leq A_s && \forall s \in \mathcal{S} && (37)\\
& \sum_{p\in\mathcal{P}} x_{pd} \geq \sum_{p\in\mathcal{P}} x_{pd'} && \forall d = 2,\dots,|\mathcal{D}|-1; d < d' \leq |\mathcal{D}| && (38)\\
& \sum_{d\in\mathcal{D}} x_{pd} \geq \sum_{p\in\mathcal{D}} x_{p'd} && \forall p \in (\mathcal{P} : s_p = s_{p'}, T_p = T_p^{'}); p < p' \leq |\mathcal{P}| && (39)\\
& x_{p1} = 1 && \forall p \in \mathcal{P}_1 && (40)\\
& x_{pd} \in \{0,1\} && \forall p \in \mathcal{P};\ d \in \mathcal{D} && (41)\\
& y_{jd} \in \mathbb{Z}^+ && \forall j \in \mathcal{J};\ d \in \mathcal{D} && (42)
\end{aligned}
$$

The RMP incorporates a bin-packing relaxation tailored for multiple surgical specialties (Constraints (34) and (35)), which is a relaxation of Constraints (17) and (18) of the MP. Instead of considering the individual capacity of each OR, Constraint (34) aggregates the individual capacity of each OR and assigns patients to the aggregated capacity of ORs. The division of assigned surgical load to each specialty by the availability time of each OR (B) results in a lower bound on the number of ORs for each specialty. Constraint (35) ensures that the number of ORs allocated to different specialties on each day does not exceed the maximum number of ORs ($|\mathcal{R}|$) in that day. Constraints (36) and (37) ensure that no surgeon's workload exceeds his/her maximum daily and weekly availability times, respectively. We also include a set of symmetry-breaking constraints, similar to those within the MP, represented by Constraints (38) and (39). Constraint (40) ensures all patients $p \in \mathcal{P}_1$ are operated on the first day of the week.

3.3.2 Subproblems

Unlike the uni-level B&C that includes only one type of subproblem (sequencing), the bi-level B&C has two types of subproblems: *packing* and *sequencing*. The RMP assigns a set of patients ($\hat{\mathcal{P}}_{jd}^{(i)}$) and ORs ($\hat{\mathcal{R}}_{jd}^{(i)}$) from a set of specialties on each day. The PSP is then an integer program responsible for allocating patients to ORs and determining whether $\hat{\mathcal{P}}_{jd}^{(i)}$ is feasibly packable given $|\hat{\mathcal{R}}_{jd}^{(i)}|$ for each specialty $j \in \hat{\mathcal{J}}_d^{(i)}$. A PSP is solved for each specialty group, $\hat{\mathcal{J}}_d^{(i)} \in \tilde{\mathcal{J}}_d^{(i)}$, that has been allocated more than one OR. The PSP model is detailed by objective function and Constraints (43)-(45):

$$\begin{aligned}
\underset{\mathbf{x}}{\text{minimize}} \quad & \mathbf{0} && && \text{(PSP)}\\
\text{subject to} \quad & \sum_{r \in \hat{\mathcal{R}}_{jd}^{(i)}} x_{pr} = 1 && \forall j \in \hat{\mathcal{J}}_d^{(i)}; p \in \hat{\mathcal{P}}_{jd}^{(i)} && (43)\\
& \sum_{p \in \hat{\mathcal{P}}_{jd}^{(i)}} (T_p + E) x_{pr} \leq B + E && \forall j \in \hat{\mathcal{J}}_d^{(i)}; r \in \hat{\mathcal{R}}_{jd}^{(i)} && (44)\\
& x_{pr} \in \{0, 1\} && \forall j \in \hat{\mathcal{J}}_d^{(i)}; p \in \hat{\mathcal{P}}_{jd}^{(i)};\ r \in \hat{\mathcal{R}}_{jd}^{(i)} && (45)
\end{aligned}$$

Constraint (43) ensures that each patient is assigned to exactly one OR, while Constraint (44) ensures that no OR is over-capacitated.

Secondary Subproblem (Sequencing). Once the RMP incumbent passes all of the PSP checks, the solution is fed into the SSPs for surgery sequencing. Optimality is proven in a similar fashion to the uni-level approach, when the lower bound of the RMP is equal to the value of the best incumbent solution that has passed all of the PSP and SSP checks.

3.3.3 Cuts

We add a Benders cut for specialty groups whose RMP allocation in incumbent i is infeasible with respect to the PSPs; we denote this set of specialty groups $\bar{\mathcal{K}}_d^{(i)}$. We use the same Benders cut for both the PSP and SSP in the bi-level B&C methods. As described in Section 3.3, the RMP provides input to the PSP and then to the SSP. We first assume the PSP does not exist and the RMP provides an incumbent directly to the SSP. If the incumbent is infeasible with respect to the SSP, the following Benders cut is added to the RMP:

$$\left(\left|\bigcup_{j \in \hat{\mathcal{J}}_d^{(i)}} \hat{\mathcal{P}}_{jd}^{(i)}\right| - \sum_{j \in \hat{\mathcal{J}}_d^{(i)}} \sum_{p \in \hat{\mathcal{P}}_{jd}^{(i)}} x_{pd}\right) + \left(\sum_{j \in \hat{\mathcal{J}}_d^{(i)}} y_{jd} - \left|\bigcup_{j \in \hat{\mathcal{J}}_d^{(i)}} \hat{\mathcal{R}}_{jd}^{(i)}\right|\right) \geq 1 \qquad \forall \hat{\mathcal{J}}_d^{(i)} \in \bar{\mathcal{K}}_d^{(i)} \qquad (46)$$

This is the author's version of an article that has been accepted for publication in the *International Journal of Production Economics*. Changes were made to this version by the publisher prior to publication. The final version of record is available at: https://doi.org/10.1016/j.ijpe.2019.07.006

To break the SSP infeasibility, this cut ensures the RMP opens at least one new OR and/or removes at least one patient from $\bigcup_{j\in\hat{\mathcal{J}}_d^{(i)}} \hat{\mathcal{P}}_{jd}^{(i)}$. The value of variable y_{jd} does not change in the future iteration if $\left|\bigcup_{j\in\hat{\mathcal{J}}_d^{(i)}} \hat{\mathcal{R}}_{jd}^{(i)}\right| = |\mathcal{R}|$ and the cut is hence reduced to the following no-good cut that can break the SSP infeasibility by only removing at least one patient from the $\hat{\mathcal{P}}_{jd}^{(i)}$, i.e., $\left(\left|\bigcup_{j\in\hat{\mathcal{J}}_d^{(i)}} \hat{\mathcal{P}}_{jd}^{(i)}\right| - \sum_{j\in\hat{\mathcal{J}}_d^{(i)}} \sum_{p\in\hat{\mathcal{P}}_{jd}^{(i)}} x_{pd}\right) \geq 1.$

Theorem 2. *Inequality (46) is a valid Benders cut. The proof is similar to that for Theorem 1.*

We now assume the PSP exists and the RMP provides incumbents to the PSP and then from the PSP to the SSP. Evidently, the RMP and the SSP can jointly define a complete algorithm on their own (similar to the uni-level approach). However, the role of the PSP is to refine the RMP incumbents with respect to the packing constraints, yielding RMP solutions that are more likely to satisfy the SSP constraints. The inclusion of the PSP in the bi-level method necessitates deriving a Benders cut for each infeasible PSP. This cut (Inequality 46), when derived from the SSP, communicates more information to the RMP regarding the correction of $\hat{\mathcal{P}}_{jd}^{(i)}$ and $\hat{\mathcal{R}}_{jd}^{(i)}$ compared to when it is derived from the PSP because the SSP includes all the constraints of the IP-MLORPS for each specialty on each day and can therefore provide more accurate feedback on the current RMP solution.

3.4 Existing Decompositions for MLORPS

A previously proposed decomposition for MLORPS (Castro and Marques, 2015) is based on Generalized Disjunctive Programming (GDP) and, similar to our uni-level B&C, decomposes the problem into a planning phase and a sequencing phase in which the surgery-to-day and surgery-to-OR assignments are checked for sequence feasibility. Upon detecting an infeasible assignment of surgeries, at least one low-priority patient is heuristically removed to ensure feasibility. Information from the sequencing phase is never communicated back to the planning phase, yielding an approximate uni-directional decomposition. The Benders cuts used in our B&C approaches remedy the approximate nature of the GDP decomposition by iteratively allowing the MP to find another allocation of patients to different days and ORs. The allocation or removal of patients determined by the MP is not conducted heuristically and the algorithm is guaranteed to converge to global optimality. The other advantage of our B&C approaches are their relative simplicity, making them more accessible for real-world implementation: the GDP reformulation incorporates 12 types of binary variables and six types of continuous variables, whereas our decompositions use slight variations of the variables from the original IP formulation.

4 Data

We consider two benchmark datasets for our empirical investigation. The first consists of instances from Marques et al. (2012) and subsequent works from the same author, while the second benchmark is new and generated in accordance to a multi-specialty variation of the case mix planning classification scheme proposed in Leeftink and Hans (2018).

4.1 Existing Benchmark

The existing benchmark uses data from Marques et al. (2012) consisting of instances with 250, 300, 500, and 1000 conventional patients drawn from the surgical wait list of a Portuguese hospital. In addition to these conventional patients, each instance also includes a number of ambulatory patients (275 on average). These instances have horizons of 4-5 days, five surgical specialties, six ORs per day, and both outpatients (12.5% of the total surgeries) and inpatients (87.5%). The availability time of each OR is 11.5 hours (46 time slots of 15 minutes) per day, and surgeon daily and weekly availability times are homogeneous and restricted to 6 hours (24 time slots) and 25 hours (100 time slots), respectively. Cleaning time is 30 minutes for all surgeries (two time slots). For modeling purposes, OR availability times are extended to 12 hours (48 time slots) to account for the cleaning time of the last surgery of the day, however, the nominal OR time is used when calculating utilization. We refer the reader to Table 2 of Castro and Marques (2015) for a complete list of parameters and their values.

In a later work, Marques et al. (2014) added new instances of 2000 conventional patients, plus additional ambulatory patients, to the aforementioned dataset. The instances provided to us, and investigated in this work, do not include these larger instances. As such, we cannot compare our proposed decomposition methods with any of the results reported on these larger 2000 conventional patient instances including the results for the GDP technique presented in Castro and Marques (2015) and some of the GA results presented in Marques et al. (2014). We note that the GA results reported in this work are taken directly from Marques et al. (2014), while both the full IP model and B&C approaches are solved using CPLEX and on the same computing platform.

4.2 New Benchmark

We create the second dataset in accordance with a recently proposed case mix planning classification scheme (Leeftink and Hans, 2018). The proposed methodology attempts to classify surgery scheduling instances according to the diversity of their underlying case mixes, measured along two axes: (i) *scheduling flexibility* (the ratio of mean surgery duration to OR capacity) and (ii) the *coefficient of variation* (the ratio of mean surgery duration to surgery duration standard deviation). Experiments on this new benchmark are intended to validate our approaches for instances with more diverse case mixes.

The generator proposed in Leeftink and Hans (2018) could not be directly applied to our problem due to the presence of predetermined surgeon-to-patient, surgeon-to-specialty, and patient priority assignments in the MLORPS problem. As such, we adopt a convenient instance generation approach that follows the methodology proposed in Leeftink and Hans (2018) closely. To construct our new benchmark, we take all the instances with 1000 conventional patients from the existing benchmark set and classify the case mix diversity of each of the five surgical specialties. To do this, we fit

This is the author's version of an article that has been accepted for publication in the *International Journal of Production Economics.* Changes were made to this version by the publisher prior to publication. The final version of record is available at: https://doi.org/10.1016/j.ijpe.2019.07.006

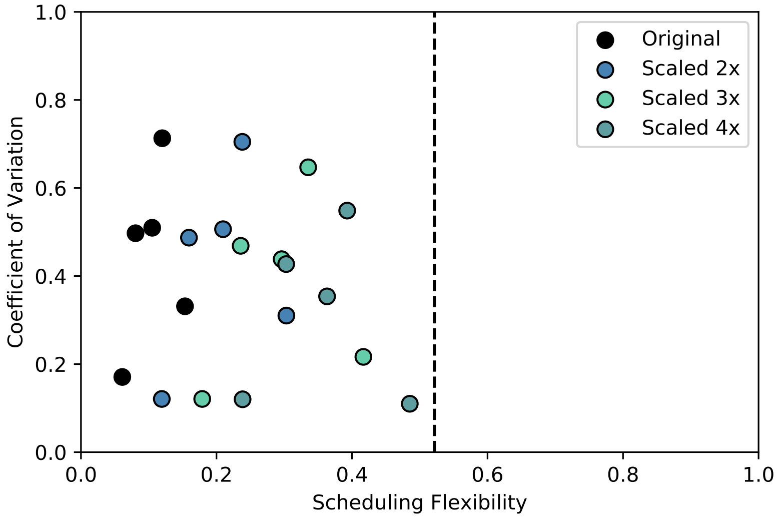

Figure 4: Case-mix plots for original and scaled surgical specialties. Each point represents the case mix diversity of the surgeries in a specialty. The vertical dashed line represents the ratio of daily surgeon availability to daily OR capacity.

a 3-parameter log-normal distribution to the surgical duration data of each specialty. We then use the parameter values from the fit to derive values for the surgical flexibility and coefficient of variation for each specialty.

The case mix diversities of each of the original five surgical specialties are illustrated in Figure 4. The original specialties have a fairly diverse coefficient of variation, but less diverse scheduling flexibility. To improve the diversity of our instance case mix, we generate new synthetic specialties by positively scaling the duration of the existing surgeries. Fifteen new specialties are generated by scaling the surgical durations of the original five specialties by two, three, and four times respectively (as illustrated by the additional points in Figure 4). We then construct our new benchmark by randomly selecting subsets of size 250, 300, 500, 700, 1000, 1500, 2000, and 3000 patients from the augmented case mix data set. For each subset size, we generate ten unique instances for a total of 80 new instances.

To help ensure the feasibility of the newly generated instances, we do not scale nor duplicate any of the deferred urgency patients nor the high priority patients. We also avoid using higher coefficients for the scaling of surgeries, as the scaled values exceed the daily availability of each surgeon, which is 360 minutes in the original data of Marques et al. (2012). Any surgeries whose scaled value exceeds the daily availability of surgeons was rounded down to 360 minutes. The dashed line in Figure 4 represents the ratio of daily surgeon availability to daily OR capacity. The case mix of each surgical specialty lies to the left of this line as only surgeries with duration less than the daily surgeon availability can be feasibly processed.

This is the author's version of an article that has been accepted for publication in the *International Journal of Production Economics*. Changes were made to this version by the publisher prior to publication. The final version of record is available at: https://doi.org/10.1016/j.ijpe.2019.07.006

5 Experimental Analysis

In this section, we experimentally validate our decomposition methods and compare them with the existing IP model (Marques et al., 2012) and GA (Marques et al., 2014) from the literature. We also explore the impact of discretization on our approaches, and provide results for our methods on the new benchmark set with more diverse case mixes.

5.1 Setup

Similar to Marques et al. (2014), we use two objective functions to compare the IP, the GA (Marques et al., 2014), and our B&C methods: i) scheduled surgical time maximization and ii) number of scheduled surgeries maximization. We consider a 30 minute time limit to compare the IP model with our B&C techniques, and two minutes to compare the GA with our B&C methods. All experiments are implemented in C++ on an 8-core machine with an Intel Core i7-6700 processor at 3.40GHz and 16GB of RAM running Red Hat Enterprise Linux, Release 6.8. We use IBM ILOG CPLEX Optimization Studio v12.6.3 for all experiments (CP Optimizer for the CP SSPs and CPLEX for the IP, RMP, MP, and PSP). We set the symmetry-breaking parameter within CPLEX to its maximum value, and leave the remainder of the parameters at their defaults. To implement the B&C methods, we utilize lazy constraint callbacks within the CPLEX solver, which are activated at each integer feasible solution to the MP and RMP, where the solution process is then directed to the associated SPs. Built-in CPLEX functions are used to evaluate solution quality and bound information throughout the search. The B&C decompositions are run single-threaded while we allow the full IP model to use all eight threads.

5.2 Existing Benchmark Instances

We compare our decomposition methods with the existing methods in the literature, namely the IP-MLORPS model solved with CPLEX (Marques et al., 2012) and the genetic algorithm (Marques et al., 2014). As the papers that presented these methods do so by discretizing surgical time into 15-minute time slots, we also discretize time in the same fashion. For these experiments we use the existing benchmark dataset.

5.2.1 IP Model

We compare our decomposition methods against IP-MLORPS solved with CPLEX. We use a 30-minute time limit for these experiments. When maximizing scheduled surgical times (in discretized time slots), the uni-level and bi-level B&C approaches without symmetry-breaking constraints both outperformed the full IP model in 100% (28 of 28) and 96% (27 of 28) of instances, yielding average optimality gaps of 1.20% and 1.21%, respectively, while the average optimality gap of the full IP model was 14.91% (Table 3). The inclusion of symmetry-breaking constraints in the uni-level approach did not improve the average optimality gap, whereas it decreased the average optimality gap in the bi-level B&C from 1.21% to 0.92%. The uni-level approach outperformed the bi-level in

This is the author's version of an article that has been accepted for publication in the *International Journal of Production Economics*. Changes were made to this version by the publisher prior to publication. The final version of record is available at: https://doi.org/10.1016/j.ijpe.2019.07.006

terms of the number of instances solved, however, the bi-level often performed better with respect to optimality gap for instances it was able to find a solution (i.e., the bi-level without SB performed better than the uni-level without SB in 20 out of 28 instances in terms of the optimality gap). The integer feasible solutions of the decomposition methods exhibit little difference, indicating the bi-level approach generates a stronger dual bound.

When maximizing the number of scheduled surgeries, the uni-level and bi-level B&C approaches without symmetry-breaking constraints outperformed the full IP model in 100% (28 of 28) and 82% (23 of 28) of instances, respectively, yielding average optimality gaps of 1.07% and 2.69%, respectively, while the average optimality gap of the IP model was 13.42% (Table 4). The uni-level approach outperformed the bi-level in terms of both the average optimality gap and number of instances solved. Symmetry-breaking constraints degraded the bi-level, solving nine fewer instances. Overall, the uni-level B&C without symmetry-breaking constraints is the best-performing method when maximizing the number of scheduled surgeries, outperforming the IP model with respect to both the solution quality and optimality gap and the bi-level method with respect to solution quality, average gap, and solvability after 30 minutes. We show in Section 5.3 that the IP model's performance depends heavily on time discretization, which is: i) less accurate for measuring OR utilization, and ii) not required for the decomposition methods.

5.2.2 Genetic Algorithm

We also compare to the performance of the GA, summarizing the results for both objective functions. For this set of experiments, we document the performance of our decompositions at two different time points: i) when a solution is found that is equal to or better than the solution found by the GA, and ii) when the decomposition algorithm has reached the runtime limit (two minutes). The former illustrates how often our B&C approaches are able to obtain solutions equal to or better than the GA given the same runtime. The latter assesses the solution quality of the algorithms given the fixed runtime limit of two minutes.

When maximizing scheduled surgical times (in discretized time slots), as shown in Table 5, the uni-level algorithm with and without symmetry-breaking constraints finds solutions equal to or better than the GA in 86% (24 of 28) of instances under both conditions in significantly less time (often 10 times less). The bi-level approach with and without symmetry-breaking constraints yields solutions equal to or better than the GA in 75% (21 of 28) and 71% (20 of 28) of instances, respectively, also in less time. Given a fixed runtime of two minutes, the uni-level B&C approach, with and without symmetry-breaking constraints, finds solutions equal to or better than the GA in 100% (28 of 28) and 96% (27 of 28) of instances, respectively. The bi-level approach, with and without symmetry-breaking constraints, yields solutions equal to or better than the GA in 86% (24 of 28) and 75% (21 of 28) of instances, respectively. On average, three out of four of our decomposition algorithms outperform the GA in terms of objective function value and computation time for each instance. The inclusion of symmetry-breaking constraints consistently improved the performance of the decomposition methods. Overall, our best decomposition, the uni-level with

This is the author's version of an article that has been accepted for publication in the *International Journal of Production Economics*. Changes were made to this version by the publisher prior to publication. The final version of record is available at: https://doi.org/10.1016/j.ijpe.2019.07.006

Table 3: IP and decomposition approaches when maximizing scheduled surgical times (in discretized time slots). All results reported after 30 minutes of runtime. Average optimality gap is reported across instances that produced a feasible solution. A dash indicates no feasible solution was found. The superscript indicates the number of instances where the decompositions outperformed the IP model. symmetry-breaking is denoted by 'SB' and bold values indicate improvement over IP.

Instance	IP		Uni-level B&C				Bi-level B&C			
			Without SB		With SB		Without SB		With SB	
	Sol	Gap	Sol	Gap	Sol	Gap	Sol	Gap	Sol	Gap
I1-250	970	12.38	**1088**	**1.71**	**1088**	**1.71**	—	—	**1092**	**0.27**
I1-300	1006	10.98	**1116**	**1.23**	**1114**	**1.39**	**1118**	**0.26**	**1117**	**0.36**
I1-500	793	32.48	**1156**	**1.54**	**1153**	**1.82**	**1158**	**0.52**	**1155**	**0.76**
I1-1000	1081	9.72	**1178**	**1.58**	**1176**	**1.79**	**1176**	**0.78**	**1176**	**0.78**
I2-250	1038	7.43	**1092**	**2.61**	**1096**	**2.25**	**1074**	**3.94**	**1072**	**4.02**
I2-300	878	22.66	**1110**	**2.23**	**1113**	**1.94**	**1096**	**3.01**	**1089**	**3.57**
I2-500	1090	7.58	**1155**	**2.02**	**1158**	**1.76**	**1112**	**5.07**	**1121**	**4.28**
I2-1000	735	38.73	**1175**	**2.05**	**1175**	**2.04**	**1150**	**3.18**	—	—
I3-250	877	20.11	**1090**	**0.69**	**1090**	**0.69**	**1091**	**0.18**	**1091**	**0.18**
I3-300	1038	7.58	**1110**	**1.15**	**1117**	**0.53**	**1117**	**0.42**	**1117**	**0.42**
I3-500	1044	10.46	**1152**	**1.18**	**1153**	**0.99**	**1159**	**0.43**	**1159**	**0.43**
I3-1000	729	39.02	**1184**	**0.94**	**1184**	**0.84**	**1178**	**1.32**	**1181**	**1.07**
I4-250	989	11.47	**1107**	**0.86**	**1110**	**0.56**	**1110**	**0.32**	**1110**	**0.32**
I4-300	1023	9.64	**1125**	**0.51**	**1126**	**0.43**	**1126**	**0.30**	**1125**	**0.39**
I4-500	1050	10.09	**1150**	**1.52**	**1160**	**0.61**	**1162**	**0.32**	**1162**	**0.32**
I4-1000	1043	13.14	**1190**	**0.86**	**1188**	**1.05**	**1189**	**0.76**	**1187**	**0.95**
I5-250	882	20.58	**1102**	**0.74**	**1103**	**0.60**	**1104**	**0.34**	**1104**	**0.34**
I5-300	956	15.14	**1118**	**0.71**	**1113**	**1.12**	**1119**	**0.34**	**1119**	**0.34**
I5-500	978	16.04	**1159**	**0.41**	**1151**	**1.10**	**1160**	**0.28**	**1159**	**0.37**
I5-1000	1032	13.87	**1189**	**0.64**	**1190**	**0.55**	**1188**	**0.61**	**1187**	**0.72**
I6-250	1044	7.55	**1118**	**0.93**	**1114**	**1.28**	**1120**	**0.12**	**1120**	**0.12**
I6-300	924	18.99	**1129**	**0.97**	**1127**	**1.13**	**1130**	**0.22**	**1130**	**0.31**
I6-500	1053	10.51	**1166**	**0.81**	**1167**	**0.70**	**1171**	**0.00**	**1171**	**0.00**
I6-1000	1101	8.29	**1183**	**1.38**	**1183**	**1.34**	**1109**	**7.06**	**1174**	**1.56**
I7-250	906	1.83	**909**	**1.27**	**910**	**1.16**	**914**	**0.44**	**914**	**0.47**
I7-300	899	2.99	**915**	**0.96**	**911**	**1.40**	**916**	**0.43**	**916**	**0.58**
I7-500	872	8.20	**938**	**0.87**	**933**	**1.40**	**935**	**0.94**	**938**	**0.61**
I7-1000	678	29.95	**951**	**1.35**	**949**	**1.56**	**950**	**1.16**	**948**	**1.37**
Avg.	953.9	14.91	1109.1	$1.20^{(28)}$	1109.0	$1.20^{(28)}$	1104.9	$1.21^{(27)}$	1105.0	$0.92^{(27)}$

This is the author's version of an article that has been accepted for publication in the *International Journal of Production Economics*. Changes were made to this version by the publisher prior to publication. The final version of record is available at: https://doi.org/10.1016/j.ijpe.2019.07.006

Table 4: IP and decomposition approaches when maximizing number of scheduled surgeries. All results reported after 30 minutes of runtime. Average optimality gap is reported across instances that produced a feasible solution. A dash indicates no feasible solution was found. The superscript indicates the number of instances where the decompositions outperformed the IP model. Symmetry-breaking is denoted by 'SB' and bold values indicate improvement over IP.

Instance	IP		Uni-level B&C				Bi-level B&C			
			Without SB		With SB		Without SB		With SB	
	Sol	Gap	Sol	Gap	Sol	Gap	Sol	Gap	Sol	Gap
I1-250	237	18.65	**282**	**0.59**	**282**	**0.59**	**282**	**2.05**	**282**	**2.05**
I1-300	248	15.01	**282**	**0.73**	**282**	**0.73**	—	—	**281**	**2.45**
I1-500	251	14.28	**285**	**0.81**	**284**	**1.15**	**282**	**2.69**	—	—
I1-1000	251	16.67	**293**	**1.98**	**295**	**1.07**	—	—	**296**	**0.60**
I2-250	251	13.39	**282**	**0.90**	**282**	**0.90**	**281**	**2.75**	**278**	**3.78**
I2-300	257	11.81	**285**	**0.78**	**284**	**1.13**	**284**	**2.41**	**281**	**3.44**
I2-500	254	13.19	**287**	**1.37**	**289**	**0.64**	**284**	**2.94**	**281**	**3.96**
I2-1000	264	12.18	**290**	**3.44**	—	—	**293**	**2.33**	—	—
I3-250	245	16.61	**285**	**0.44**	**285**	**0.44**	**280**	**4.07**	**282**	**3.36**
I3-300	276	6.19	**285**	**0.95**	**285**	**0.76**	—	—	—	—
I3-500	249	16.05	**288**	**1.66**	**289**	**1.13**	**283**	**4.00**	**286**	**2.99**
I3-1000	253	16.94	**300**	**0.92**	**301**	**0.59**	**297**	**1.92**	—	—
I4-250	257	13.64	**288**	**0.78**	**287**	**1.13**	**265**	**10.05**	**288**	**2.24**
I4-300	282	5.37	**288**	**1.04**	**288**	**1.77**	**288**	**2.37**	279	5.42
I4-500	250	16.83	**294**	**1.21**	**293**	**1.58**	**293**	**1.55**	**294**	**1.21**
I4-1000	258	16.40	**305**	**0.00**	**301**	**1.51**	**301**	**1.51**	—	—
I5-250	282	5.11	**287**	**0.87**	**287**	**0.63**	—	—	—	—
I5-300	269	9.61	**288**	**1.18**	**288**	**0.91**	**288**	**2.08**	**286**	**2.92**
I5-500	248	17.33	**291**	**2.02**	**292**	**1.68**	**291**	**2.02**	**293**	**1.35**
I5-1000	251	18.45	**304**	**0.00**	**301**	**1.25**	**300**	**1.57**	—	—
I6-250	249	15.65	**287**	**0.62**	**287**	**0.62**	**287**	**2.05**	**285**	**3.00**
I6-300	267	9.61	**287**	**0.96**	**287**	**0.97**	**287**	**2.33**	—	—
I6-500	239	19.47	**290**	**1.97**	**292**	**0.55**	**288**	**2.37**	—	—
I6-1000	249	18.25	**301**	**0.81**	**300**	**1.06**	—	—	—	—
I7-250	234	3.54	**236**	**1.11**	**236**	**1.13**	**236**	**2.88**	—	—
I7-300	237	2.63	**238**	**0.80**	**238**	**0.57**	**238**	**2.06**	—	—
I7-500	201	17.83	**241**	**1.25**	**241**	**1.15**	**240**	**1.88**	**240**	**1.88**
I7-1000	215	14.95	**251**	**0.71**	**252**	**0.00**	**247**	**1.98**	—	—
Avg.	250.9	13.42	282.9	$1.07^{(28)}$	282.5	$0.95^{(27)}$	278.9	$2.69^{(23)}$	282.1	$2.71^{(14)}$

This is the author's version of an article that has been accepted for publication in the *International Journal of Production Economics*. Changes were made to this version by the publisher prior to publication. The final version of record is available at: `https://doi.org/10.1016/j.ijpe.2019.07.006`

Table 5: GA and decompositions when maximizing scheduled surgical times (in discretized time slots). Decompositions: Runtime and quality of the first solution of equal or better quality than the GA (Time and Sol_1, respectively), and quality of the best solution found in two minutes (Sol_2). A dash indicates no such solution was found in the allotted time limit. The superscript indicates the number of instances that the decompositions outperformed the GA. Symmetry-breaking is denoted by 'SB', bold values indicate an improvement over the GA, and italic numbers are the GA solutions which are better than decompositions.

Instance	GA		Uni-level B&C						Bi-level B&C					
			Without SB			With SB			Without SB			With SB		
	Time	Sol	Time	Sol_1	Sol_2	Time	Sol_1	Sol_2	Time	Sol_1	Sol_2	Time	Sol_1	Sol_2
I1-250	50.6	1075	**3.6**	**1075**	**1080**	**4.8**	**1076**	**1081**	—	—	—	**1.2**	**1083**	**1091**
I1-300	51.5	1093	**5.6**	**1093**	**1109**	**1.9**	**1100**	**1107**	**0.4**	**1096**	**1118**	**0.5**	**1100**	**1117**
I1-500	61.8	1123	**3.0**	**1137**	**1140**	**2.5**	**1131**	**1144**	**1.1**	**1149**	**1155**	**25.6**	**1146**	**1151**
I1-1000	81.5	1138	**3.4**	**1155**	**1167**	**5.3**	**1156**	**1156**	—	—	1132	—	—	1136
I2-250	51.8	1083	**1.5**	**1089**	**1092**	**1.8**	**1090**	**1096**	—	—	—	—	—	1072
I2-300	56.8	1092	**14.9**	**1100**	**1103**	**23.3**	**1099**	**1107**	—	—	—	—	—	1089
I2-500	62.4	1116	**1.3**	**1140**	**1149**	**23.9**	**1140**	**1148**	—	—	—	95.7	1121	**1121**
I2-1000	97.7	1129	**2.7**	**1161**	**1161**	**34.1**	**1156**	**1156**	—	—	1092	—	—	—
I3-250	56.3	1081	**33.8**	**1081**	**1089**	114.8	1086	**1086**	**2.8**	**1088**	**1090**	**1.7**	**1081**	**1090**
I3-300	62.3	1098	**4.3**	**1104**	**1105**	**2.2**	**1101**	**1107**	**0.6**	**1099**	**1117**	**1.0**	**1099**	**1117**
I3-500	72.0	1126	**1.9**	**1126**	**1133**	**1.8**	**1130**	**1151**	**2.6**	**1144**	**1155**	**0.3**	**1142**	**1157**
I3-1000	113.1	1141	**2.7**	**1168**	**1168**	**4.3**	**1169**	**1175**	**10.4**	**1162**	**1176**	**112.0**	**1151**	**1177**
I4-250	69.0	1094	112.2	1097	**1104**	**1.5**	**1094**	**1102**	**0.5**	**1101**	**1110**	**0.4**	**1097**	**1110**
I4-300	65.4	1109	**45.5**	**1111**	**1115**	**58.5**	**1109**	**1122**	**49.0**	**1110**	**1126**	**1.5**	**1111**	**1120**
I4-500	70.4	1130	**3.2**	**1144**	**1145**	**5.6**	**1144**	**1144**	**0.2**	**1147**	**1160**	**0.6**	**1144**	**1159**
I4-1000	116.3	1152	**5.2**	**1176**	**1176**	**4.8**	**1166**	**1166**	**8.7**	**1163**	**1174**	**7.9**	**1159**	**1179**
I5-250	60.2	1097	—	—	1091	117.6	1098	**1098**	**1.5**	**1101**	**1104**	**1.5**	**1100**	**1104**
I5-300	68.6	1108	110.1	1108	**1116**	115.3	1108	**1108**	**20.6**	**1109**	**1115**	**3.0**	**1117**	**1119**
I5-500	75.8	1133	**2.3**	**1136**	**1158**	**2.9**	**1134**	**1151**	**0.5**	**1146**	**1160**	**0.5**	**1149**	**1158**
I5-1000	109.2	1153	**2.2**	**1160**	**1180**	**55.9**	**1167**	**1167**	**7.5**	**1167**	**1184**	**2.1**	**1175**	**1187**
I6-250	50.0	1101	**10.6**	**1103**	**1109**	**8.1**	**1101**	**1110**	**1.0**	**1102**	**1117**	**0.5**	**1102**	**1120**
I6-300	46.8	1110	77.0	1120	**1126**	63.6	1116	**1126**	**0.4**	**1112**	**1129**	**0.4**	**1113**	**1127**
I6-500	71.6	1133	**3.7**	**1140**	**1162**	**3.5**	**1138**	**1151**	**1.2**	**1147**	**1169**	**0.8**	**1151**	**1169**
I6-1000	106.9	1141	**4.8**	**1161**	**1163**	**3.5**	**1160**	**1175**	—	—	1034	113.3	1150	**1153**
I7-250	35.5	894	**0.5**	**896**	**907**	**1.6**	**896**	**910**	**0.2**	**900**	**914**	**0.2**	**895**	**913**
I7-300	38.6	896	**1.3**	**896**	**913**	**0.7**	**901**	**911**	**0.9**	**908**	**916**	**0.4**	**902**	**916**
I7-500	47.4	907	**1.4**	**923**	**938**	**1.0**	**920**	**933**	**6.8**	**912**	**935**	**2.0**	**928**	**933**
I7-1000	73.4	917	**1.2**	**923**	**948**	**1.9**	**932**	**941**	110.0	937	**939**	108.7	937	**942**
Avg.	68.7	1084.6	17.0	$1093.4^{(24)}$	$1101.7^{(27)}$	23.8	$1093.5^{(24)}$	$1101.0^{(28)}$	10.8	$1085.7^{(20)}$	$1096.7^{(21)}$	20.1	$1089.7^{(21)}$	$1101.0^{(24)}$

SB, finds feasible solutions of the same quality of the GA at least six times faster when averaged over the solved instances.

When maximizing the number of scheduled surgeries, as illustrated in Table 6, the uni-level B&C approach, with and without symmetry-breaking constraints, finds solutions equal to or better than the GA in 82% (23 of 28) and 93% (26 of 28) of instances, respectively, in significantly less time than the GA. Conversely, the bi-level approach, with and without symmetry-breaking constraints, both only yield solutions equal to or better than the GA in 36% (10 of 28) of instances, respectively. The inclusion of symmetry-breaking constraints negatively impacted the uni-and bi-level decomposition methods, resulting in fewer instances being solved overall. When looking at runtime to a comparable solution to the GA (Sol_1), the uni-level decomposition without SB finds solutions six times faster when averaged over solved instances. For the two minute runtime (Sol_2), the uni-level B&C approaches with and without symmetry-breaking constraints found solutions

This is the author's version of an article that has been accepted for publication in the *International Journal of Production Economics*. Changes were made to this version by the publisher prior to publication. The final version of record is available at: https://doi.org/10.1016/j.ijpe.2019.07.006

equal to or better than the GA in 96% (27 of 28) of instances under both conditions, and the bi-level B&C approaches yield solutions equal to or better than the GA in only 43% (12 of 28) and 61% (17 of 28) of instances, respectively. There is no instance that the bi-level B&C approaches can solve that at least one of the uni-level methods cannot.

Overall, the experiments demonstrate that our exact decomposition approaches are competitive with an approximate GA metaheuristic, while providing *provable bounds* on solution quality and, often, finding solutions of equivalent quality to those found by the GA in less runtime.

5.3 Impact of Surgical Time Discretization

The experiments in the previous sections, following those of Marques et al. (2012), have discretized time into 15-minute slots (surgery durations, surgeon and OR availability). In the original problem instances, the surgical times are divided by 15 and rounded up to the nearest integer (e.g., a 256-minute procedure results in 18 time slots). This discretization may lead to sub-optimal solutions compared to the use of the exact surgical durations. To investigate the impact of time discretization, we re-run the experiments based the original (non-discretized) surgical times.[1] Before doing so, we examine the impact of different time slot sizes on the performance of the IP-MLORPS model solved with CPLEX (Table 7). It is evident that the model cannot be used for time slot discretization of less than 10 minutes for even the smallest instance. Furthermore, when the original surgical times are used without discretization, CPLEX is unable to formulate the model. As the size of the time slots is increased the performance improves, until CPLEX is eventually able to solve the root node relaxation for the formulation based on 5-minute time slots. This issue becomes even more apparent for problem instances larger than the instance assessed in Table 7, indicating the need for a technique that scales more effectively for non-discretized data.

Table 8 illustrates the computational results of the various methods when maximizing non-discretized scheduled surgical time. The uni-level approaches solve all instances of the problem, but the bi-level approaches, with and without symmetry-breaking constraints, could solve only 96% (27 out of 28) and 86% (24 out of 28) instances of the problem, respectively. In general, the average optimality gap of all B&C approaches using non-discretized surgical times appears to be similar to the discretized experiments. The IP-MLORPS model was unable to solve even the smallest problems, largely due to memory issues as detailed in Table 7.

When maximizing the number of scheduled surgeries with exact data, as detailed in Table 9, the uni-level B&C approaches, with and without symmetry-breaking constraints, were able to solve 100% (28 out of 28) of instances, whereas the bi-level B&C approaches with and without symmetry-breaking constraints only solved 35% (10 out of 28) and 64% (18 out of 28) of the problem, respectively. The optimality gaps of the uni-level approach are in the same range as for the discretized experiments. Similarly, the IP-MLORPS model was unable to solve any of the problems.

We can conclude from the obtained results in Tables 8 and 9 that the uni-level B&C approaches

[1]'Non-discretized' means the durations are expressed in their original integer minute format.

This is the author's version of an article that has been accepted for publication in the *International Journal of Production Economics*. Changes were made to this version by the publisher prior to publication. The final version of record is available at: `https://doi.org/10.1016/j.ijpe.2019.07.006`

Table 6: GA and decompositions when maximizing the number of scheduled surgeries. Decompositions: Runtime and quality of the first solution of equal or better quality than the GA (Time and Sol_1, respectively), and quality of the best solution found in two minutes (Sol_2). A dash indicates no such solution was found in the allotted time limit. The superscript indicates the number of instances that decompositions outperformed the GA. Symmetry-breaking is denoted by 'SB', bold values indicate an improvement over the GA, and italic numbers are the GA solutions which are better than decompositions.

Instance	GA		Uni-level B&C						Bi-level B&C					
			Without SB			With SB			Without SB			With SB		
	Time	Sol	Time	Sol_1	Sol_2	Time	Sol_1	Sol_2	Time	Sol_1	Sol_2	Time	Sol_1	Sol_2
I1-250	38.0	242	**1.3**	**269**	**282**	**2.3**	**251**	**282**	**6.4**	**281**	**282**	**11.9**	**282**	**282**
I1-300	38.2	255	**1.2**	**270**	**281**	**4.1**	**273**	**281**	—	—	—	**19.9**	**281**	**281**
I1-500	28.8	270	**15.8**	**282**	**284**	**16.2**	**280**	**284**	66.5	280	**280**	—	—	—
I1-1000	36.4	287	42.0	289	**290**	81.6	289	**289**	—	—	—	**2.1**	**291**	**294**
I2-250	43.6	235	**4.2**	**259**	**282**	**25.2**	**267**	**281**	—	—	—	—	—	—
I2-300	33.1	246	**1.0**	**263**	**284**	**10.2**	**258**	**284**	—	—	—	**5.3**	**279**	**281**
I2-500	33.0	269	**2.9**	**273**	**287**	**3.1**	**274**	**286**	—	—	—	118.9	278	**278**
I2-1000	30.7	285	—	—	—	—	—	—	—	—	—	—	—	—
I3-250	32.9	248	**1.2**	**272**	**285**	**1.7**	**283**	**284**	112.1	262	**263**	**26.6**	**263**	**267**
I3-300	48.2	252	**1.1**	**258**	**285**	**4.5**	**277**	**285**	—	—	—	—	—	—
I3-500	32.1	271	**2.3**	**281**	**288**	**7.0**	**284**	**289**	110.1	274	**283**	**22.7**	**284**	**286**
I3-1000	34.3	287	**6.5**	**287**	**298**	63.9	291	**296**	—	—	262	—	—	—
I4-250	31.7	245	**0.9**	**253**	**288**	**3.9**	**280**	**287**	112.2	263	**265**	118.5	284	—
I4-300	31.4	254	**1.4**	**280**	**288**	**6.5**	**282**	**288**	**19.1**	**287**	**288**	110.6	276	**279**
I4-500	44.8	272	**2.1**	**276**	**294**	**8.6**	**286**	**293**	**23.6**	**291**	**292**	**9.5**	**293**	**294**
I4-1000	37.4	290	**24.1**	**300**	**301**	74.0	296	**299**	53.5	301	**301**	—	—	—
I5-250	38.1	248	**1.6**	**261**	**287**	**5.7**	**274**	**287**	—	—	—	—	—	—
I5-300	26.6	259	**1.9**	**268**	**288**	**5.7**	**278**	**288**	—	—	—	—	—	—
I5-500	42.2	275	**3.0**	**276**	**290**	**4.9**	**275**	**292**	**0.7**	**279**	**291**	**2.2**	**291**	**292**
I5-1000	35.4	289	**6.1**	**295**	**299**	75.5	296	**296**	82.9	298	**299**	—	—	—
I6-250	42.7	240	**3.1**	**248**	**287**	**6.0**	**279**	**286**	**14.8**	**283**	**287**	**13.4**	**285**	**285**
I6-300	34.7	251	**1.9**	**266**	**287**	**7.9**	**265**	**287**	35.6	282	**287**	—	—	—
I6-500	34.5	266	**2.8**	**278**	**290**	**2.9**	**281**	**292**	**21.8**	**286**	**288**	—	—	—
I6-1000	36.5	283	**5.8**	**290**	**299**	**7.4**	**295**	**295**	—	—	—	—	—	—
I7-250	44.3	205	**0.8**	**227**	**236**	**1.7**	**225**	**236**	**21.6**	**233**	**236**	—	—	—
I7-300	37.8	213	**1.4**	**213**	**238**	**2.4**	**231**	**237**	**18.1**	**236**	**238**	—	—	—
I7-500	35.7	226	**1.3**	**234**	**241**	**3.5**	**236**	**241**	**12.2**	**238**	**240**	**29.5**	**236**	**240**
I7-1000	31.5	239	**18.6**	**247**	**251**	**4.5**	**244**	**244**	**28.9**	**246**	**246**	—	—	—
Avg.	36.2	257.2	5.8	$267.2^{(26)}$	$281.9^{(27)}$	16.3	$272.2^{(23)}$	$281.1^{(27)}$	43.5	$271.8^{(10)}$	$273.8^{(17)}$	37.8	$278.7^{(10)}$	$279.9^{(12)}$

This is the author's version of an article that has been accepted for publication in the *International Journal of Production Economics*. Changes were made to this version by the publisher prior to publication. The final version of record is available at: https://doi.org/10.1016/j.ijpe.2019.07.006

Table 7: The impact of time discretization on the performance of IP-MLORPS implemented in CPLEX. Runtimes are expressed in seconds and the experiment has been done for the smallest instance of the problem (I-250). 'O.O.M:' indicates out of memory

Time discretization	Formulate model	Presolve (first pass)	Presolve incumbent	Root node relaxation
1-minute	O.O.M	—	—	—
2-minute	84.2	O.O.M	—	—
3-minute	11.1	139.9	O.O.M	—
4-minute	6.7	68.5	82.7	O.O.M
5-minute	4.8	37.2	47.7	783.3
10-minute	1.7	10.7	13.5	58.1
15-minute	1.1	6.3	7.9	25.9

are the most effective for solving MLORPS with exact surgical duration data. Interestingly, using exact data, instead of the discretized time slots, often gave a boost to the performance of the uni-level B&C approach. A possible reason for this could be the natural symmetry-breaking that occurs when using the exact data. The process of discretizing time into time slots often results in procedures with unequal original durations being treated as the same duration (e.g., procedures with durations of 18 and 20, after discretization, both have a duration of two time slots), which increases the symmetry in the problem.

Using non-discretized data also has significant implications on the objective function values. Figure 5 illustrates the average objective value of the uni-level decomposition with symmetry-breaking for both time discretized (i.e., 15 minute time slots) and non-discretized experiments. It is clear from the figure that as the number of patients considered for optimization increases, both objective functions obtain better values. This finding is independent of the way time is discretized. However, time discretization has a mixed impact on the values of the objective functions. For instance, discretization results in an *overestimation* when maximizing scheduled surgical times (discussed also in Castro and Marques (2015)), but leads to an *underestimation* when maximizing the number of scheduled surgeries. The solutions obtained from the discretized data are not, therefore, the true optimal values for the objective functions.

Consider an example instance consisting of four surgeries of the same surgical specialty with surgical durations of 196, 211, 166, and 19 minutes, a single OR (with associated availability of 690 minutes, or 46 discretized time slots), a single surgeon, and a mandatory cleaning time of 30 minutes after each surgery (two time slots). Converting these surgeries to discretized time slots of 15 minutes yields: 14, 15, 12, and 2 time slots, respectively. Maximizing scheduled surgical times will result in the first three procedures being scheduled, as there is not enough available time for all four, with an associated utilization of 89.1% (41/46 time slots used, where cleaning time is not part of the utilization calculation).[2] The same objective with non-discretized data allows the scheduling of all the surgeries and yields an OR utilization of 85.8% (592/690). This example illustrates how the discretized experiments can show higher scheduled surgical times while scheduling fewer

[2]If time slots are scaled back into minutes before calculating utilization, via multiplication by 15, the same conclusion holds.

This is the author's version of an article that has been accepted for publication in the *International Journal of Production Economics*. Changes were made to this version by the publisher prior to publication. The final version of record is available at: https://doi.org/10.1016/j.ijpe.2019.07.006

Table 8: Decompositions when maximizing scheduled surgical times (in minutes). A runtime of 30 minutes is used. IP did not find any integer solution. The average optimality gap is for solved instances. Superscript: The total number of best integer feasible solutions found and the total number of instances with lowest optimality gap achieved by each B&C method. SB: symmetry-breaking. —: No integer feasible solution found. Bold: Best objective function value and optimality gap in each instance.

Instance	Uni-level B&C				Bi-level B&C			
	Without SB		With SB		Without SB		With SB	
	Sol	Gap	Sol	Gap	Sol	Gap	Sol	Gap
I1-250	**15945**	**1.91**	15809	2.74	15228	6.00	—	—
I1-300	16354	1.83	16357	1.81	—	—	**16436**	**0.79**
I1-500	14626	16.14	17142	1.66	16354	5.50	**17177**	**0.82**
I1-1000	17460	**1.89**	**17464**	1.91	16329	7.43	16509	6.44
I2-250	16244	1.58	16173	2.00	**16343**	**0.55**	16288	0.95
I2-300	16441	1.79	16273	2.77	16441	1.37	**16520**	**0.73**
I2-500	**17221**	1.60	17181	1.83	17205	**0.99**	17162	1.27
I2-1000	**17529**	**1.61**	17513	1.74	16640	5.74	—	—
I3-250	**16002**	**0.59**	15881	1.35	14268	11.34	15944	0.81
I3-300	16245	1.84	16234	1.90	**16440**	**0.61**	16420	0.65
I3-500	17022	1.32	16944	1.77	16917	1.92	**17130**	**0.68**
I3-1000	17598	**0.80**	**17599**	**0.80**	16895	4.69	16595	6.39
I4-250	16052	2.18	16256	0.95	**16261**	**0.52**	16235	0.72
I4-300	16443	1.28	16477	1.15	16288	2.13	**16524**	**0.71**
I4-500	17085	1.09	17022	1.43	**17136**	**0.71**	17126	0.77
I4-1000	17596	1.32	**17654**	**0.93**	16829	5.51	16834	5.49
I5-250	16191	0.83	16153	1.06	**16232**	**0.28**	16206	0.46
I5-300	16424	0.90	**16431**	**0.82**	15985	3.46	16211	2.11
I5-500	17038	1.19	16969	1.59	**17081**	**0.87**	17068	0.96
I5-1000	17604	1.04	**17613**	**1.01**	16735	5.86	16488	7.26
I6-250	16286	1.82	16379	1.26	16376	0.78	**16393**	**0.66**
I6-300	16519	1.65	**16579**	1.30	16468	1.52	16537	**1.13**
I6-500	17128	1.63	17188	1.28	**17205**	**0.78**	17035	1.75
I6-1000	17546	1.62	**17560**	**1.54**	16249	8.31	—	—
I7-250	13360	1.48	13341	1.62	**13497**	**0.44**	13490	0.49
I7-300	13434	1.51	13477	1.19	13532	0.62	**13546**	**0.52**
I7-500	**13885**	**0.95**	13875	1.03	13235	5.55	12964	7.50
I7-1000	**14193**	**0.89**	**14193**	**0.89**	13634	4.70	—	—
Avg.	$16266.8^{(6)}$	$1.94^{(7)}$	$16347.8^{(8)}$	$1.48^{(6)}$	$15992.7^{(8)}$	$3.27^{(9)}$	$16201.6^{(7)}$	$2.09^{(8)}$

This is the author's version of an article that has been accepted for publication in the *International Journal of Production Economics*. Changes were made to this version by the publisher prior to publication. The final version of record is available at: https://doi.org/10.1016/j.ijpe.2019.07.006

Table 9: Decompositions when maximizing number of scheduled surgeries (in non-discretized minutes). A runtime of 30 minutes is used. IP did not find any integer solution. The average optimality gap is for solved instances. Superscript: The total number of best integer feasible solutions found and the total number of instances with lowest optimality gap achieved by each B&C method. SB: symmetry-breaking. —: No integer feasible solution found. Bold: Best objective function value and optimality gap in each instance.

Instance	Uni-level B&C				Bi-level B&C			
	Without SB		With SB		Without SB		With SB	
	Sol	Gap	Sol	Gap	Sol	Gap	Sol	Gap
I1-250	**302**	1.67	**302**	**1.47**	—	—	299	4.27
I1-300	**306**	**0.76**	305	1.86	—	—	—	—
I1-500	306	3.26	**309**	**0.91**	—	—	—	—
I1-1000	319	3.41	**322**	**2.20**	—	—	319	3.71
I2-250	302	2.43	**303**	**1.22**	300	3.78	302	3.08
I2-300	305	**1.52**	**306**	1.82	300	4.71	**306**	2.81
I2-500	310	1.80	**311**	**1.35**	—	—	—	—
I2-1000	320	3.42	**322**	**2.75**	—	—	—	—
I3-250	**306**	0.91	**306**	**0.87**	—	—	**306**	3.13
I3-300	**307**	2.72	**307**	**1.20**	—	—	304	4.22
I3-500	**313**	2.50	312	**1.82**	—	—	310	4.15
I3-1000	**323**	**3.39**	**323**	**3.39**	304	9.06	320	4.29
I4-250	**310**	0.90	**310**	**0.69**	—	—	309	3.10
I4-300	**311**	**0.80**	309	1.44	—	—	309	3.37
I4-500	**317**	**3.11**	316	3.32	—	—	—	—
I4-1000	326	3.69	**331**	**1.95**	243	28.19	—	—
I5-250	309	**1.16**	309	1.20	—	—	**310**	2.61
I5-300	311	1.99	**312**	**0.63**	—	—	310	2.95
I5-500	**316**	**1.14**	315	3.03	—	—	315	3.36
I5-1000	328	2.14	**331**	**1.86**	329	2.30	—	—
I6-250	**308**	**1.29**	307	2.47	305	3.31	**308**	2.40
I6-300	309	1.29	**310**	**0.86**	—	—	—	—
I6-500	313	2.84	**314**	**1.59**	**314**	2.72	308	4.58
I6-1000	**325**	**2.53**	323	2.77	312	6.43	—	—
I7-250	254	1.78	**255**	**1.54**	252	4.81	254	4.05
I7-300	**257**	**1.29**	**257**	1.35	—	—	256	3.81
I7-500	**262**	**0.92**	261	1.34	—	—	259	4.37
I7-1000	**273**	**1.99**	272	2.21	268	4.36	—	—
Avg.	$305.3^{(15)}$	$2.02^{(12)}$	$305.7^{(18)}$	$1.75^{(17)}$	$292.7^{(1)}$	$6.97^{(0)}$	$300.2^{(4)}$	$3.57^{(0)}$

This is the author's version of an article that has been accepted for publication in the *International Journal of Production Economics*. Changes were made to this version by the publisher prior to publication. The final version of record is available at: https://doi.org/10.1016/j.ijpe.2019.07.006

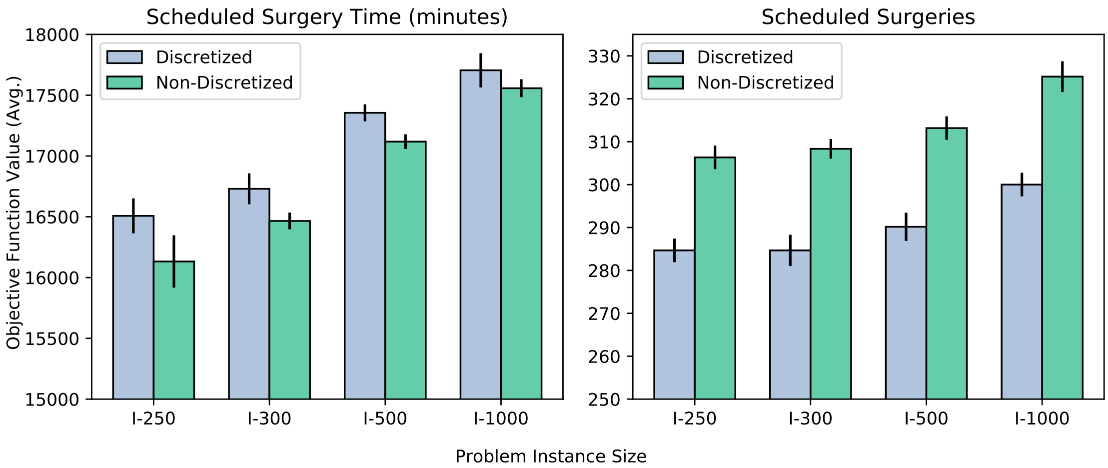

Figure 5: Impact of surgical time discretization on the objective function values. The average value is taken for solved trials using the uni-level approach with symmetry-breaking. For the maximization of scheduled surgical times, discretized results are scaled by 15 to convert them to values in minutes. The vertical line on top of each bar represents the standard deviation.

surgeries than the non-discretized variant, reflecting the results in Figure 5. It also indicates that the utilization conclusions of discretized experiments must be interpreted with care, and that our uni-level decomposition, with good performance on non-discretized data, provides a more accurate model of the utilization of hospital OR resources.

5.4 New Benchmark Instances

In this section, we extend our performance evaluation to validate our approaches on more diverse case mix benchmark instances (see Section 4.2). To do so, we only appraise the performance of the uni-level B&C with symmetry-breaking constraints, which outperformed the other algorithms for both objective functions in the non-discretized experiments.

Our newly-generated benchmark consists of 80 non-discretized instances of varying sizes, including those larger than in Marques et al. (2012). The uni-level B&C method is able to find integer feasible solutions for all of problem instances on both objective functions (Tables 10 and 11). Given that the time-indexed IP model solved with CPLEX struggles to find any solutions for non-discretized data, as shown in the previous section, the results are not included. The average optimality gap is slightly higher overall than those reported for the original benchmark set for both objective functions. As our instance generation scheme does not place any requirements on the number of deferred urgency nor high priority patients, the increased presence of optional surgeries (the dominating class of surgeries in the data set) may result in deteriorated relaxation strength. Additionally, the presence of more surgical specialties in the new data set could lead to more fragmented subproblems. The standard deviation is relatively consistent across the instance sizes. Interestingly, the performance improves on both objective functions as we increase the size of the problem; as the pool of available surgeries grows, solutions with higher objective function

This is the author's version of an article that has been accepted for publication in the *International Journal of Production Economics*. Changes were made to this version by the publisher prior to publication. The final version of record is available at: https://doi.org/10.1016/j.ijpe.2019.07.006

Table 10: The performance of the uni-level B&C with symmetry-breaking constraints when maximizing scheduled surgical times on the newly generated non-discretized dataset. A runtime of 30 minutes is used. Standard deviation, min, and max are reported for the optimality gaps of the associated instance size.

Instance Size	# Feasible	Avg. Gap	Std. Dev.	Min	Max
250	10	3.18	0.49	2.51	4.05
300	10	2.92	0.24	2.53	3.27
500	10	2.84	0.30	2.33	3.29
700	10	2.70	0.26	2.17	2.97
1000	10	2.04	0.31	1.44	2.41
1500	10	1.77	0.35	1.38	2.60
2000	10	1.42	0.26	1.14	1.80
3000	10	1.32	0.35	0.87	1.84
Average	10	2.27	0.32	1.80	2.78

values are more likely. As the objective function value increases, with a proptionally increasing bound, the optimality gap will trend smaller.

Given the uni-level B&C's average optimality gaps of 2.27% and 3.45% on non-discretized data, comparable to the results obtained for the non-discretized original benchmark with poor case-mix diversity, we can conclude that the approach is suitable for solving instances with more diverse case mixes as well.

Table 11: The performance of the uni-level B&C with symmetry-breaking constraints when maximizing number of scheduled surgeries on the newly generated non-discretized dataset. A runtime of 30 minutes is used. Standard deviation, min, and max are reported for the optimality gaps of the associated instance size.

Instance Size	# Feasible	Avg. Gap	Std. Dev.	Min	Max
250	10	5.70	1.16	3.65	7.22
300	10	5.84	0.96	4.31	7.40
500	10	4.22	0.54	3.39	5.07
700	10	3.23	0.38	2.66	3.98
1000	10	2.36	0.27	1.90	2.77
1500	10	2.40	0.39	1.75	2.85
2000	10	2.02	0.57	1.12	2.75
3000	10	1.81	0.48	0.98	2.50
Average	10	3.45	0.59	2.47	4.32

6 Discussion

We divide our discussion into two sections: methodology and application.

6.1 Methodology

The uni-level approaches demonstrated the strongest performance among the tested methods. While the uni-level B&C method was able to find solutions to nearly all the instances for both objective functions, the bi-level approach exhibited predominantly poor performance, particularly on the exact (non-discretized) variant of MLORPS.

The poor performance of the bi-level B&C approach for MLORPS is in contrast to the existing literature that reports significant computational savings obtained by a bi-level LBBD compared to a uni-level LBBD for a real-world OR scheduling problem (Riise et al., 2016). For a stochastic facility location/fleet management problem, Fazel-Zarandi et al. (2013) demonstrated that the poor performance of their bi-level LBBD was likely attributable to the amount of information that was pushed out of the MP compared to uni-level LBBD. Unlike Fazel-Zarandi et al. (2013), we postulate that the poor performance of our bi-level approach is due to the existence of a large number of symmetrical solutions. Beck (2010) proposed a variant of B&C, coined OPT15, that limits SP checking to incumbents with optimality gap of $\leq 15\%$, but this approach is not expected to help our decomposition approaches because they converge to $\leq 15\%$ optimality gap extremely fast in almost all cases, although future work could investigate a tighter threshold for our problem.

Metaheuristics, including the GA compared against in this paper, are commonly perceived to be ideal solution candidates for solving large-scale deterministic and stochastic problems. Nevertheless, we showed that our B&C approaches can outperform the GA using the discretized data. The quality of GA solutions are usually assessed against the bounds or integer solutions of the mathematical models developed for the same problem. We showed that for the non-discretized data, the IP model could not even be constructed for the smallest size of the problem given a 30-minute runtime. Therefore, to calculate the optimality gap of the GA solutions and to assess the GA quality of integer solutions, one can compare them with the best bound and the best integer solution obtained by our proposed decompositions on each instance, respectively. Our decompositions do not suffer from this weakness and can provide bounds via the LP relaxations attained within the MP search.

6.2 Application

When maximizing the scheduled surgical time, surgeries with longer duration are favored. The maximum utilization schedule has one surgery in each OR whose surgical time spans the daily availability time of each OR (a case that rarely occurs in reality). Alternatively, when maximizing the number of scheduled surgeries, shorter surgeries are preferred, resulting in a large total cleaning time and poor OR utilization. *Maximization of scheduled surgical times is suitable for keeping resources in ORs highly occupied, whereas maximizing the number of scheduled surgeries shortens wait lists.*

We showed that the performance of our uni-level B&C is not sensitive to the way data is discretized, whereas the IP model is extremely sensitive to the data discretization. To achieve a trade-off between computational tractability and accurate objective values, the exact techniques in OR scheduling have been applied to one-minute (Riise et al., 2016; Roshanaei et al., 2017a,b), five-

This is the author's version of an article that has been accepted for publication in the *International Journal of Production Economics*. Changes were made to this version by the publisher prior to publication. The final version of record is available at: https://doi.org/10.1016/j.ijpe.2019.07.006

minute (Hashemi Doulabi et al., 2016), and 15-minute (Marques et al., 2012; Vijayakumar et al., 2013) time slot discretization. We demonstrated that our B&C approaches are both efficient and accurate, indicating future time discretization for OR scheduling is unnecessary. However, we also note that OR scheduling with discretized surgical durations (assuming the discretization rounds the exact value up) may actually result in more robust schedules due to the added duration slack. Following the work of Davenport et al. (2014), we plan to study slack-based techniques for creating robust OR schedules.

The choice of objective function has other clinical implications on downstream units (e.g., post-anesthesia care unit, intensive care unit, ward unit, step-down unit). Insufficient downstream beds may cause up to 18% elective patient cancellations (Wang et al., 2016). Since MLORPS does not consider downstream capacities, it might overestimate the number of surgeries that can be scheduled. OR schedules constructed to maximize scheduled surgical times are likely to yield under-utilized downstream beds, whereas OR schedules maximizing the volume of surgeries are more susceptible to surgery cancellations due to downstream capacity limits. An additional factor that is of clinical importance is the way OR times are distributed among specialties. In the vast majority of OR scheduling applications, ORs are either exclusively reserved for a surgeon (block scheduling) or for surgeons of a certain specialty (open scheduling). This constraint has been incorporated into almost all the mathematical models focusing OR allocation to multiple specialties (Castro and Marques, 2015; Marques et al., 2012, 2014). Relaxing this constraint and scheduling surgeries of multiple specialties within an OR has been shown to result in long turnover times (up two twice as large) (Marques et al., 2012).

7 Conclusions and future work

We developed novel uni-level and bi-level branch-and-check (B&C) methods for multi-level operating room planning and scheduling. We compared our B&C methods with CPLEX on an existing IP model and a genetic algorithm with surgical times discretized into 15-minute time slots. Given a two minute runtime limit, the B&C methods outperform the GA in terms of computational time and solution quality. The uni-level approach demonstrated the strongest overall performance among B&C methods and found solutions better than the GA in 86% and 93% of instances on objective functions of scheduled surgical times maximization and number of surgeries maximization, respectively, providing significant computational savings. When compared to CPLEX on an existing IP model with a 30-minute runtime, uni-level B&C improved the optimality gap of the IP model by at least 10 times. Our decomposition methods, using the same 30-minute runtime, performed equally well on non-discretized surgical times, whereas the IP model exceeded available memory, even for the smallest instance. We demonstrated that the use of discretized data in MLORPS overestimates OR utilization and underestimates the number of surgeries that can be scheduled. Therefore, the solutions obtained from discretized data are not true optima with respect to both objective functions. To further evaluate the applicability of our B&C methods, we generated a new dataset consisting of up to 3000 patients and 20 surgical specialties, and showed that the average

This is the author's version of an article that has been accepted for publication in the *International Journal of Production Economics*. Changes were made to this version by the publisher prior to publication. The final version of record is available at: https://doi.org/10.1016/j.ijpe.2019.07.006
optimality gaps of our best B&C method are 2.27% and 3.45% when we maximize the scheduled surgical times and the number of scheduled surgeries, respectively.

As part of our future work, we plan to expand the problem definition to include other elements, such as multiple hospitals, multiple stages (pre-operative and post-operative in addition to intra-operative) of an operating theatre, and the sharing of OR resources among surgical specialties. Each of these new elements is accompanied with modeling and algorithmic challenges, likely requiring more complex decompositions (e.g., sequence dependent cleaning times for specialties sharing ORs) and algorithmic improvements, such as Benders cut propagation (Roshanaei et al., 2017a) or proliferation (Heching et al., 2019). We also plan on investigating the application of our methods to stochastic problem variants that involve some level of uncertainty; as noted briefly before, the notion of adding slack to surgical durations is one option here (Davenport et al., 2014).

This is the author's version of an article that has been accepted for publication in the *International Journal of Production Economics*. Changes were made to this version by the publisher prior to publication. The final version of record is available at: https://doi.org/10.1016/j.ijpe.2019.07.006

A Proof of Theorem 1

Proof. We first show that Inequality (31) rules out the MP incumbent i ($\hat{x}^{(i)}_{pdr}$ and $\hat{y}^{(i)}_{jdr}$) from the feasible set (Property 1). Let $\bigcup_{j\in\hat{\mathcal{J}}^{(i)}_d} \hat{\mathcal{P}}^{(i)}_{jd}$ and $\bigcup_{j\in\hat{\mathcal{J}}^{(i)}_d} \hat{\mathcal{R}}^{(i)}_{jd}$ be the current sets of patients and ORs, respectively, assigned to specialties $j \in \hat{\mathcal{J}}^{(i)}_d$, leading to an infeasible SSP. We denote by $\bar{\mathcal{J}}^{(i)}_d$ the set of infeasible $\hat{\mathcal{J}}^{(i)}_d$. By definition, $\hat{\mathcal{P}}^{(i)}_{jd} = \{p \in \mathcal{P}_j, d \in \mathcal{D}, r \in \mathcal{R} \mid x_{pdr} = 1\}$ and $\hat{\mathcal{R}}^{(i)}_{jd} = \{j \in \mathcal{J}, d \in \mathcal{D}; r \in \mathcal{R} \mid y_{jdr} = 1 \text{ for MP incumbent } i\} \neq \emptyset$. To show that Inequality (31) rules out the current MP solution, we instantiate it with $\bigcup_{j\in\hat{\mathcal{J}}^{(i)}_d} \hat{\mathcal{P}}^{(i)}_{jd}$ and $\bigcup_{j\in\hat{\mathcal{J}}^{(i)}_d} \hat{\mathcal{R}}^{(i)}_{jd}$ yielding:

$$\left(\overbrace{\left|\bigcup_{j\in\hat{\mathcal{J}}^{(i)}_d} \hat{\mathcal{P}}^{(i)}_{jd}\right| - \sum_{j\in\hat{\mathcal{J}}^{(i)}_d}\sum_{p\in\hat{\mathcal{P}}^{(i)}_{jd}}\sum_{r\in\hat{\mathcal{R}}^{(i)}_{jd}} x_{pdr}}^{=0}\right) + \left(\overbrace{\sum_{j\in\hat{\mathcal{J}}^{(i)}_d}\sum_{r\in\mathcal{R}} y_{jdr} - \left|\bigcup_{j\in\hat{\mathcal{J}}^{(i)}_d} \hat{\mathcal{R}}^{(i)}_{jd}\right|}^{=0}\right) \ngeq (<)1 \qquad \forall \hat{\mathcal{J}}^{(i)}_d \in \bar{\mathcal{J}}^{(i)}_d \tag{47}$$

indicating that the current MP solution does not satisfy Inequality (47) and it is therefore eliminated from the MP feasible region.

We now demonstrate that future feasible solutions satisfy Inequality (31) (Property 2). Consider a hypothetical future solution ($\tilde{x}_{pdr}, \tilde{y}_{jdr}$), leading to a new set of patients and ORs allocated to specialties $j \in \hat{\mathcal{J}}^{(i)}_d$ on day d, denoted by $\bigcup_{j\in\hat{\mathcal{J}}^{(i)}_d} \tilde{\mathcal{P}}^{(i)}_{jd}$ and $\bigcup_{j\in\hat{\mathcal{J}}^{(i)}_d} \tilde{\mathcal{R}}^{(i)}_{jd}$, respectively. We first assume that the hypothetical future solution keeps the same selection of ORs determined by the MP at iteration i (i.e., $\left|\bigcup_{j\in\hat{\mathcal{J}}^{(i)}_d} \tilde{\mathcal{R}}^{(i)}_{jd}\right| = \left|\bigcup_{j\in\hat{\mathcal{J}}^{(i)}_d} \hat{\mathcal{R}}^{(i)}_{jd}\right|$) and only makes changes to the set of patients in $\bigcup_{j\in\hat{\mathcal{J}}^{(i)}_d} \tilde{\mathcal{P}}^{(i)}_{jd}$. We expand the first part of Inequality (47) (for each $\hat{\mathcal{J}}^{(i)}_d \in \bar{\mathcal{J}}^{(i)}_d$) as follows:

$$\left(\overbrace{\left|\bigcup_{j\in\hat{\mathcal{J}}^{(i)}_d} \hat{\mathcal{P}}^{(i)}_{jd} \bigcap \bigcup_{j\in\hat{\mathcal{J}}^{(i)}_d} \tilde{\mathcal{P}}^{(i)}_{jd}\right| - \sum_{j\in\hat{\mathcal{J}}^{(i)}_d}\sum_{p\in\hat{\mathcal{P}}^{(i)}_{jd}\cap\tilde{\mathcal{P}}_{jd}}\sum_{r\in\hat{\mathcal{R}}^{(i)}_{jd}} \tilde{x}_{pdr}}^{a}\right) + \left(\overbrace{\left|\bigcup_{j\in\hat{\mathcal{J}}^{(i)}_d} \hat{\mathcal{P}}^{(i)}_{jd} \setminus \bigcup_{j\in\hat{\mathcal{J}}^{(i)}_d} \tilde{\mathcal{P}}^{(i)}_{jd}\right| - \sum_{j\in\hat{\mathcal{J}}^{(i)}_d}\sum_{p\in\hat{\mathcal{P}}^{(i)}_{jd}\setminus\tilde{\mathcal{P}}_{jd}}\sum_{r\in\hat{\mathcal{R}}^{(i)}_{jd}} \tilde{x}_{pdr}}^{b}\right) \geq 1. \tag{48}$$

a is by definition 0, but b can be ≥ 1 (feasible future solution) or < 1 (infeasible future solution). Therefore, to show that the future feasible solution satisfies Inequality (48), we just have to show $b \geq 1$. There exist four possible scenarios between $\bigcup_{j\in\hat{\mathcal{J}}^{(i)}_d} \hat{\mathcal{P}}^{(i)}_{jd}$ and $\bigcup_{j\in\hat{\mathcal{J}}^{(i)}_d} \tilde{\mathcal{P}}^{(i)}_{jd}$, the first three of which satisfy Inequality (48):

This is the author's version of an article that has been accepted for publication in the *International Journal of Production Economics*. Changes were made to this version by the publisher prior to publication. The final version of record is available at: https://doi.org/10.1016/j.ijpe.2019.07.006

1. $\bigcup_{j\in\hat{\mathcal{J}}_d^{(i)}} \tilde{\mathcal{P}}_{jd}^{(i)}$ subtracts some patients from $\bigcup_{j\in\hat{\mathcal{J}}_d^{(i)}} \hat{\mathcal{P}}_{jd}^{(i)}$ (i.e., $\bigcup_{j\in\hat{\mathcal{J}}_d^{(i)}} \hat{\mathcal{P}}_{jd}^{(i)} \supset \bigcup_{j\in\hat{\mathcal{J}}_d^{(i)}} \tilde{\mathcal{P}}_{jd}^{(i)}$ and $\bigcup_{j\in\hat{\mathcal{J}}_d^{(i)}} \hat{\mathcal{P}}_{jd}^{(i)} \setminus \bigcup_{j\in\hat{\mathcal{J}}_d^{(i)}} \tilde{\mathcal{P}}_{jd}^{(i)} \neq \emptyset$), yielding $b \geq 1$ because $\tilde{x}_{pdr} = 0$ for $\bigcup_{j\in\hat{\mathcal{J}}_d^{(i)}} \hat{\mathcal{P}}_{jd}^{(i)} \setminus \bigcup_{j\in\hat{\mathcal{J}}_d^{(i)}} \tilde{\mathcal{P}}_{jd}^{(i)}$. Therefore, $\bigcup_{j\in\hat{\mathcal{J}}_d^{(i)}} \tilde{\mathcal{P}}_{jd}^{(i)}$ satisfies Inequality (48).

2. $\bigcup_{j\in\hat{\mathcal{J}}_d^{(i)}} \tilde{\mathcal{P}}_{jd}^{(i)}$ shares some patients with $\bigcup_{j\in\hat{\mathcal{J}}_d^{(i)}} \hat{\mathcal{P}}_{jd}^{(i)}$ (i.e., $\bigcup_{j\in\hat{\mathcal{J}}_d^{(i)}} \hat{\mathcal{P}}_{jd}^{(i)} \bigcap \bigcup_{j\in\hat{\mathcal{J}}_d^{(i)}} \tilde{\mathcal{P}}_{jd}^{(i)} \neq \emptyset$), yielding $b \geq 1$ because $\tilde{x}_{pdr} = 0$ for $\bigcup_{j\in\hat{\mathcal{J}}_d^{(i)}} \hat{\mathcal{P}}_{jd}^{(i)} \setminus \bigcup_{j\in\hat{\mathcal{J}}_d^{(i)}} \tilde{\mathcal{P}}_{jd}^{(i)}$. Therefore, $\bigcup_{j\in\hat{\mathcal{J}}_d^{(i)}} \tilde{\mathcal{P}}_{jd}^{(i)}$ satisfies Inequality (48).

3. $\bigcup_{j\in\hat{\mathcal{J}}_d^{(i)}} \tilde{\mathcal{P}}_{jd}^{(i)}$ shares no patient with $\bigcup_{j\in\hat{\mathcal{J}}_d^{(i)}} \hat{\mathcal{P}}_{jd}^{(i)}$ (i.e., $\bigcup_{j\in\hat{\mathcal{J}}_d^{(i)}} \hat{\mathcal{P}}_{jd}^{(i)} \bigcap \bigcup_{j\in\hat{\mathcal{J}}_d^{(i)}} \tilde{\mathcal{P}}_{jd}^{(i)} = \emptyset$), yielding $b \geq 1$ because $\tilde{x}_{pdr} = 0$ for $\bigcup_{j\in\hat{\mathcal{J}}_d^{(i)}} \hat{\mathcal{P}}_{jd}^{(i)} \setminus \bigcup_{j\in\hat{\mathcal{J}}_d^{(i)}} \tilde{\mathcal{P}}_{jd}^{(i)}$. Therefore, $\bigcup_{j\in\hat{\mathcal{J}}_d^{(i)}} \tilde{\mathcal{P}}_{jd}^{(i)}$ satisfies Inequality (48).

4. $\bigcup_{j\in\hat{\mathcal{J}}_d^{(i)}} \tilde{\mathcal{P}}_{jd}^{(i)}$ adds some new patients to $\bigcup_{j\in\hat{\mathcal{J}}_d^{(i)}} \hat{\mathcal{P}}_{jd}^{(i)}$ (i.e., $\bigcup_{j\in\hat{\mathcal{J}}_d^{(i)}} \hat{\mathcal{P}}_{jd}^{(i)} \subset \bigcup_{j\in\tilde{\mathcal{J}}_d^{(i)}} \tilde{\mathcal{P}}_{jd}^{(i)}$ and $\bigcup_{j\in\hat{\mathcal{J}}_d^{(i)}} \hat{\mathcal{P}}_{jd}^{(i)} \setminus \bigcup_{j\in\hat{\mathcal{J}}_d^{(i)}} \tilde{\mathcal{P}}_{jd}^{(i)} = \emptyset$), yielding $b = 0 < 1$. Therefore, $\bigcup_{j\in\hat{\mathcal{J}}_d^{(i)}} \tilde{\mathcal{P}}_{jd}^{(i)}$ is infeasible, which is intuitive because $\bigcup_{j\in\hat{\mathcal{J}}_d^{(i)}} \tilde{\mathcal{P}}_{jd}^{(i)}$ has more patients than $\bigcup_{j\in\hat{\mathcal{J}}_d^{(i)}} \hat{\mathcal{P}}_{jd}^{(i)}$, while $\bigcup_{j\in\hat{\mathcal{J}}_d^{(i)}} \hat{\mathcal{P}}_{jd}^{(i)}$ is already infeasible. Therefore, $\bigcup_{j\in\hat{\mathcal{J}}_d^{(i)}} \tilde{\mathcal{P}}_{jd}^{(i)}$ does not satisfy Inequality (48).

Unlike Scenarios 1-3, Scenario 4 results in a future solution that is a superset of the current MP solution, which does not satisfy Inequality (48).

We repeat the same procedure for ORs. We alternatively assume that the set of patients in the hypothetical future solution, $\bigcup_{j\in\hat{\mathcal{J}}_d^{(i)}} \tilde{\mathcal{P}}_{jd}^{(i)}$, is identical to $\bigcup_{j\in\hat{\mathcal{J}}_d^{(i)}} \hat{\mathcal{P}}_{jd}^{(i)}$, but the set of ORs allocated to specialty j on day d will change. By keeping the same set of patients in the future solution, Inequality (31) is reduced to (for each $\hat{\mathcal{J}}_d^{(i)} \in \bar{\mathcal{J}}_d^{(i)}$):

$$\sum_{j\in\hat{\mathcal{J}}_d^{(i)}} \sum_{r\in\mathcal{R}} y_{jdr} - \left| \bigcup_{j\in\hat{\mathcal{J}}_d^{(i)}} \hat{\mathcal{R}}_{jd}^{(i)} \right| \geq 1. \tag{49}$$

We first show that the future MP solution, $\tilde{y}_{jdr}$, that allocates the same number of ORs to the infeasible specialty groups $\bar{\mathcal{J}}_d^{(i)}$ ($\left|\bigcup_{j\in\hat{\mathcal{J}}_d^{(i)}} \hat{\mathcal{R}}_{jd}^{(i)}\right|$) does not satisfy:

$$\sum_{j\in\hat{\mathcal{J}}_d^{(i)}} \sum_{r\in\mathcal{R}} y_{jdr} - \left| \bigcup_{j\in\hat{\mathcal{J}}_d^{(i)}} \hat{\mathcal{R}}_{jd}^{(i)} \right| \ngeq (<)1. \tag{50}$$

Cut (49) is satisfied by those MP solutions that increase $\left|\bigcup_{j\in\hat{\mathcal{J}}_d^{(i)}} \hat{\mathcal{R}}_{jd}^{(i)}\right|$ by one least one OR.

To show that Inequality (31) satisfies all feasible future solutions due to simultaneous changes

This is the author's version of an article that has been accepted for publication in the *International Journal of Production Economics*. Changes were made to this version by the publisher prior to publication. The final version of record is available at: https://doi.org/10.1016/j.ijpe.2019.07.006
in $\bigcup_{j\in\hat{\mathcal{J}}_d^{(i)}}\hat{\mathcal{P}}_{jd}^{(i)}$ and $\left|\bigcup_{j\in\hat{\mathcal{J}}_d^{(i)}}\hat{\mathcal{R}}_{jd}^{(i)}\right|$, we reformulate Inequality (31) as

$$\left(\left|\bigcup_{j\in\hat{\mathcal{J}}_d^{(i)}}\hat{\mathcal{P}}_{jd}^{(i)}\bigcap\bigcup_{j\in\hat{\mathcal{J}}_d^{(i)}}\tilde{\mathcal{P}}_{jd}^{(i)}\right|-\sum_{j\in\hat{\mathcal{J}}_d^{(i)}}\sum_{p\in\hat{\mathcal{P}}_{jd}^{(i)}\cap\tilde{\mathcal{P}}_{jd}}\sum_{r\in\hat{\mathcal{R}}_{jd}^{(i)}}\tilde{x}_{pdr}\right)+$$
$$\left(\left|\bigcup_{j\in\hat{\mathcal{J}}_d^{(i)}}\hat{\mathcal{P}}_{jd}^{(i)}\setminus\bigcup_{j\in\hat{\mathcal{J}}_d^{(i)}}\tilde{\mathcal{P}}_{jd}^{(i)}\right|-\sum_{j\in\hat{\mathcal{J}}_d^{(i)}}\sum_{p\in\hat{\mathcal{P}}_{jd}^{(i)}\setminus\tilde{\mathcal{P}}_{jd}}\sum_{r\in\hat{\mathcal{R}}_{jd}^{(i)}}\tilde{x}_{pdr}\right)+$$
$$\left(\sum_{j\in\hat{\mathcal{J}}_d^{(i)}}\sum_{r\in\mathcal{R}}\tilde{y}_{jdr}-\left|\bigcup_{j\in\hat{\mathcal{J}}_d^{(i)}}\hat{\mathcal{R}}_{jd}^{(i)}\right|\right)\geq 1,\qquad\forall\hat{\mathcal{J}}_d^{(i)}\in\bar{\mathcal{J}}_d^{(i)}\tag{51}$$

For all the feasible scenarios, Inequality (51) will take a value ≥ 1, and thus no feasible solutions are ruled out. For the sake of completeness, we show the following *infeasible* future solutions $\bigcup_{j\in\hat{\mathcal{J}}_d^{(i)}}\hat{\mathcal{P}}_{jd}^{(i)}\subseteq\bigcup_{j\in\hat{\mathcal{J}}_d^{(i)}}\tilde{\mathcal{P}}_{jd}^{(i)}$ while $\left|\bigcup_{j\in\hat{\mathcal{J}}_d^{(i)}}\hat{\mathcal{R}}_{jd}^{(i)}\right|=\left|\bigcup_{j\in\hat{\mathcal{J}}_d^{(i)}}\tilde{\mathcal{R}}_{jd}^{(i)}\right|$ and $\bigcup_{j\in\hat{\mathcal{J}}_d^{(i)}}\hat{\mathcal{P}}_{jd}^{(i)}=\bigcup_{j\in\hat{\mathcal{J}}_d^{(i)}}\tilde{\mathcal{P}}_{jd}^{(i)}$ while $\left|\bigcup_{j\in\hat{\mathcal{J}}_d^{(i)}}\hat{\mathcal{R}}_{jd}^{(i)}\right|\geq\left|\bigcup_{j\in\hat{\mathcal{J}}_d^{(i)}}\tilde{\mathcal{R}}_{jd}^{(i)}\right|$ do not satisfy Inequality (51):

$$\left(\overbrace{\left|\bigcup_{j\in\hat{\mathcal{J}}_d^{(i)}}\hat{\mathcal{P}}_{jd}^{(i)}\bigcap\bigcup_{j\in\hat{\mathcal{J}}_d^{(i)}}\tilde{\mathcal{P}}_{jd}^{(i)}\right|-\sum_{j\in\hat{\mathcal{J}}_d^{(i)}}\sum_{p\in\hat{\mathcal{P}}_{jd}^{(i)}\cap\tilde{\mathcal{P}}_{jd}}\sum_{r\in\hat{\mathcal{R}}_{jd}^{(i)}}\tilde{x}_{pdr}}^{=0}\right)+$$
$$\left(\overbrace{\left|\bigcup_{j\in\hat{\mathcal{J}}_d^{(i)}}\hat{\mathcal{P}}_{jd}^{(i)}\setminus\bigcup_{j\in\hat{\mathcal{J}}_d^{(i)}}\tilde{\mathcal{P}}_{jd}^{(i)}\right|-\sum_{j\in\hat{\mathcal{J}}_d^{(i)}}\sum_{p\in\hat{\mathcal{P}}_{jd}^{(i)}\setminus\tilde{\mathcal{P}}_{jd}}\sum_{r\in\hat{\mathcal{R}}_{jd}^{(i)}}\tilde{x}_{pdr}}^{=0}\right)+$$
$$\left(\overbrace{\left(\sum_{j\in\hat{\mathcal{J}}_d^{(i)}}\sum_{r\in\mathcal{R}}\tilde{y}_{jdr}-\left|\bigcup_{j\in\hat{\mathcal{J}}_d^{(i)}}\hat{\mathcal{R}}_{jd}^{(i)}\right|\right)}^{=0}\right)\ngeq 1,\qquad\forall\hat{\mathcal{J}}_d^{(i)}\in\bar{\mathcal{J}}_d^{(i)}\tag{52}$$

□

B Proof of Theorem 2

Proof. The proof is similar to Theorem 1. □

References

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

Aringhieri, Roberto, Paolo Landa, Patrick Soriano, Elena Tànfani, Angela Testi. 2015. A two level metaheuristic for the operating room scheduling and assignment problem. *Computers & Operations Research* **54** 21 – 34.

Augusto, V., X. Xie, V. Perdomo. 2010. Operating theatre scheduling with patient recovery in both operating rooms and recovery beds. *Computers & Industrial Engineering* **58**(2) 231–238.

Banditori, C., P. Cappanera, F. Visintin. 2013. A combined optimization-simulation approach to the master surgical scheduling problem. *IMA Journal of Management Mathematics* **24**(2) 155–187.

Baptiste, C., C. Le Pape, W. Nuijten. 2001. *Constraint-based Scheduling*. Kluwer Academic Publishers.

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

Batun, S., B. T. Denton, T. R. Huschka, A. J. Schaefer. 2011. Operating room pooling and parallel surgery processing under uncertainty. *INFORMS Journal on Computing* **23**(2) 220–237.

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

Belien, J., E. Demeulemeester. 2008. A branch-and-price approach for integrating nurse and surgery scheduling. *European Journal of Operational Research* **189**(3) 652–668.

Booth, K. E. C., T. T. Tran, J. C. Beck. 2016. Logic-based Benders decomposition methods for the travelling purchaser problem. Claude-Guy Quimper, ed., *Principles and Practice of Constraint Programming - CP 2016*, *Lecture Notes in Computer Science*, vol. 5732. Springer Berlin Heidelberg, 55–64.

Cardoen, B., E. Demeulemeester, J. Belien. 2009. Sequencing surgical cases in a day-care environment: An exact branch-and-price approach. *Computers & Operations Research* **36**(9) 2660–2669.

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

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

Davenport, Andrew, Christophe Gefflot, Chris Beck. 2014. Slack-based techniques for robust schedules. *Sixth European Conference on Planning*.

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

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

Fazel-Zarandi, M. M., J. C. Beck. 2012. Using logic-based Benders decomposition to solve the capacity- and distance-constrained plant location problem. *INFORMS Journal on Computing* **24**(3) 387–398.

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

Fei, H., C. Chu, N. Meskens. 2009. Solving a tactical operating room planning problem by a column-generation-based heuristic procedure with four criteria. *Annals of Operations Research* **166**(1) 91–108.

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

Fei, H., N. Meskens, C. Chu. 2010. A planning and scheduling problem for an operating theatre using an open scheduling strategy. *Computers & Industrial Engineering* **58**(2) 221–230.

Gedik, R., C. Rainwater, H. Nachtmann, Ed A. Pohl. 2016. Analysis of a parallel machine scheduling problem with sequence dependent setup times and job availability intervals. *European Journal of Operational Research* **251**(2) 640–650.

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

Harjunkoski, I., I. E. Grossmann. 2002. Decomposition techniques for multistage scheduling problems using mixed-integer and constraint programming methods. *Computers & Chemical Engineering* **26**(11) 1533–1552.

Hashemi Doulabi, S. H., L. M. Rousseau, G. Pesant. 2016. A constraint-programming-based branch-and-price-and-cut approach for operating room planning and scheduling. *INFORMS Journal on Computing* **28**(3) 432–448.

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

HFMA. 2005. Achieving operating room efficiency through process integration (Health Care Financial Management Association Report).

Hooker, J. N. 1994. Logic-based methods for optimization. Alan Borning, ed., *Principles and Practice of Constraint Programming*, *Lecture Notes in Computer Science*, vol. 874. Springer Berlin Heidelberg, 336–349.

Hooker, J. N. 2007. Planning and scheduling by logic-based Benders decomposition. *Operations Research* **55**(3) 588–602.

Hooker, J. N., G. Ottosson. 2003. Logic-based Benders decomposition. *Mathematical Programming* **96**(1) 33–60.

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

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

Leeftink, G., E. W. Hans. 2018. Case mix classification and a benchmark set for surgery scheduling. *Journal of Scheduling* **21**(1) 17–33.

Marques, I. 2010. Planeamento de Cirurgias Electivas-Abordagens em Programacao Inteira. *Ph.D. Thesis, Faculdade de Ciencias, Universidade de Lisboa (in Portuguese)* .

Marques, I., M. E. Captivo. 2015. Bicriteria elective surgery scheduling using an evolutionary algorithm. *Operations Research for Health Care* **7** 14–26.

Marques, I., M. E. Captivo, V. P. Margarida. 2012. An integer programming approach to elective surgery scheduling. *OR Spectrum* **34**(2) 407–427. doi:10.1007/s00291-011-0279-7.

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

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

Pham, D. N., A. Klinkert. 2008. Surgical case scheduling as a generalized job shop scheduling problem. *European Journal of Operational Research* **185**(3) 1011–1025.

Rath, S., K. Rajaram, A. Mahajan. 2017. Integrated anesthesiologist and room scheduling for surgeries: methodology and application. *Operations Research* **65**(6) 1460–1478.

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

Roshanaei, V., C. Luong, D. Aleman, D. Urbach. 2017a. Propagating logic-based Benders' decomposition approaches for distributed operating room scheduling. *European Journal of Operational Research* **257**(2) 439–455.

Roshanaei, V., C. Luong, D. M. Aleman, D. Urbach. 2019. Reformulation, linearization, and decomposition techniques for balanced distributed operating room scheduling. *Omega* doi: https://doi.org/10.1016/j.omega.2019.03.001. URL `http://www.sciencedirect.com/science/article/pii/S0305048317309015`.

Roshanaei, V., C. Luong, M. Aleman, D. Urbach. 2017b. Collaborative operating room planning and scheduling. *INFORMS Journal on Computing* **29**(3) 558–580.

Samudra, M., C. Van Riet, E. Demeulemeester, B. Cardoen, N. Vansteenkiste, F. E. Rademakers. 2016. Scheduling operating rooms: achievements, challenges and pitfalls. *Journal of Scheduling* **19**(5) 493–525.

Shylo, O. V., O. A. Prokopyev, A. J. Schaefer. 2013. Stochastic operating room scheduling for high-volume specialties under block booking. *INFORMS Journal on Computing* **25**(4) 682–692. doi:10.1287/ijoc.1120.0530.

Silva, T. A. O., M. C. de Souza, R. R. Saldanha, E. K. Burke. 2015. Surgical scheduling with simultaneous employment of specialised human resources. *European Journal of Operational Research* **245**(3) 719 – 730.

Tanfani, E., A. Testi. 2010. A pre-assignment heuristic algorithm for the Master Surgical Schedule Problem (MSSP). *Annals of Operations Research* **178**(1) 105–119.

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

Tran, T. T, A. Araujo, J. C. Beck. 2016. Decomposition methods for the parallel machine scheduling problem with setups. *INFORMS Journal on Computing* **28**(1) 83–95.

Van Riet, C., E. Demeulemeester. 2015. Trade-offs in operating room planning for electives and emergencies: A review. *Operations Research for Health Care* **7** 52 – 69.

Vijayakumar, B., P. J. Parikh, R. Scott, A. Barnes, J. Gallimore. 2013. A dual bin-packing approach to scheduling surgical cases at a publicly-funded hospital. *European Journal of Operational Research* **224**(3) 583 – 591.

Vissers, J., I. Adan, J. Bekkers. 2005. Patient mix optimization in cardiothoracic surgery planning: a case study. *IMA J Manag Math* **16**(3) 281–304.

Wang, S., V. Roshanaei, D. Aleman, D. Urbach. 2016. A discrete event simulation evaluation of distributed operating room scheduling. *IIE Transactions on Healthcare Systems Engineering* **6**(4) 236–245.

Wang, T., N. Meskens, D. Duvivier. 2015. Scheduling operating theatres: Mixed integer programming vs. constraint programming. *European Journal of Operational Research* **247**(2) 401 – 413.

Zhu, S., W. Fan, S. Yang, J. Pei, P.M. Paradalos. 2018. Operating room planning and surgical case scheduling: a review of literature. *Journal of Combinatorial Optimization* doi:10.1007/s10878-018-0322-6.