



HAL
open science

A branch-and-price-and-cut algorithm for operating room scheduling under human resource constraints

Roberto Baretto, Thierry Garaix, Xiaolan Xie

► **To cite this version:**

Roberto Baretto, Thierry Garaix, Xiaolan Xie. A branch-and-price-and-cut algorithm for operating room scheduling under human resource constraints. *Computers and Operations Research*, 2023, 152, pp.106136. 10.1016/j.cor.2022.106136 . hal-04609352

HAL Id: hal-04609352

<https://hal.science/hal-04609352v1>

Submitted on 12 Jun 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A branch-and-price-and-cut algorithm for operating room scheduling under human resource constraints

Roberto Baretto^{a,*}, Thierry Garaix¹, Xiaolan Xie^{1,1}

^aCenter for Biomedical & Health Care Engineering, École des Mines de Saint-Étienne, Saint-Étienne, France and LIMOS UMR CNRS 6158, 158 cours Fauriel, 42023 Cedex 2, Saint-Étienne, France

Abstract

In this paper, we address an integrated operating room planning and scheduling problem that includes, with fine detail, constraints commonly encountered in practice (*i.e.*, sequence, capacity and due date constraints) and for human resources other than surgeons, *i.e.*, nurses. A new model of the sequence-dependent operating room cleaning times that arise because of surgeries with different infection levels is considered. To solve this difficult integrated planning and scheduling problem, we devise a branch-and-price-and-cut algorithm based on the time-indexed formulation of the problem. The basic column generation scheme relies on a label-correcting algorithm that we purposely developed for solving the pricing problems that are modeled as single operating room scheduling problems with time-dependent costs and sequence-dependent cleaning times. The pricing problems are strongly NP-Hard. The efficiency of the label-correcting algorithm is ensured by dominance rules among labels and by two algorithms for computing the upper and lower bound of labels. An effective cutting procedure, inspired by Benders' decomposition and based on duality theory for linear programming, is developed for tightening the linear relaxation of the problem. With instances from the literature and that we generated, we conduct a numerical study to demonstrate the computational effectiveness of the solution method.

Keywords: operating room scheduling, operations research in healthcare, column generation, labeling algorithm, Benders' cuts, branch-and-price-and-cut

1. Introduction

In this work, we extend the integrated operating room planning and scheduling (IORPS) problem of [17] to include resource constraints for human resources (HR) other than surgeons, *i.e.*, nurses. We consider the same ILP problem modeling proposed in [17] extending it with the HR constraints for the nurses availability; moreover, we develop a new model of the sequence-dependent operating room (OR) cleaning. This optimization problem supports decisions made on a weekly basis on the schedule of surgeries for the next week. The health condition of the patients and the availability of surgeons are the main constraints involved in this decision process. However, a surgery team with specific skills

is required to conduct each surgery. In general, except the surgeon, the members of the surgery team can be decided on the fly since the required skills are shared by several operators. However the lack of operators with the required skill at the needed time may delay the operations and imply surgeries rescheduling.

To solve such a difficult problem, we devise a branch-and-price-and-cut algorithm. The column generation (CG) underlying the algorithm is based on the time-indexed MILP formulation of the problem and the pricing problem is solved through a label correcting algorithm we purposely developed. The algorithm for pricing new columns exploits the reformulation of the problem as the search of path over graph. This pricing problem reformulation is inspired by the work [31] where a similar approach is developed for solving a single machine scheduling problem through CG. The CG is speed up with dominance rules partly inspired by those

*Corresponding author

¹X. Xie is also with Shanghai Jiao Tong University, Shanghai, China.

of [17]. The more an algorithm is fast for computing surgery schedules, the more it appeals to healthcare professionals. They prefer short time computing methods to generate several schedules (or like to be able to) before fixing the final one. The need for recomputing schedules may arise for different reasons, *e.g.*, sudden changes in the list of patients. Moreover, CG is generally faster than other solution approaches to recompute surgery schedules after some data has changed.

The main contribution of this work is a novel Benders' cutting procedure that we developed for tightening the problem LP relaxation computed through CG. Such a cutting procedure is inspired by three-phase algorithm proposed in [7] where the classic Benders' decomposition is combined with CG to solve the problem of the simultaneous assignment of locomotives and cars to passenger trains. The three-phase algorithm has been applied also for the simultaneous scheduling of aircraft and crew [8, 25, 26]. The basic idea of the algorithm of [7] consists in deriving a Benders' subproblem for each resource other than that considered the master resource, *e.g.*, in [8] Benders' cuts generated solving the subproblem for scheduling crews are added to the master problem for scheduling aircraft. In this work, ORs are master resources and surgeons and nurses are slave resources. Differently from the cited literature, the Benders' cutting procedure that we devise in this paper is not the classic one, *i.e.*, the master resource solution enforced into the subproblem is partial and no constraint is dropped from the problem model of the master resources.

This work has some limitations. We do not consider uncertainty of surgery durations. Models including uncertainty and stochastic optimization methods are powerful tools to cope with uncertainty if the available information on surgery durations is weak. In this work, we assume that enough information is given for an accurate estimation of surgery durations as surgeons are already assigned to surgeries, this is shown to reduce uncertainty [19]. We consider a single linear objective for the optimization problem. The solution algorithm that we develop is tested with instances requiring to maximize the total revenue of scheduled surgeries or the total surgery time of scheduled surgeries. There are no changes to apply to the problem model or the algorithm. Only the problem data have to change to consider one of the two objectives. Our formulation of the problem objective can be

easily adapted to other objectives, *e.g.*, maximizing the total score of scheduled surgeries to prioritize urgent surgeries, *i.e.*, those with an assigned high score. Similarly, the throughput of the ORs can be maximized by assigning to every surgery a score equal to 1. For considering other optimization objectives like surgery tardiness or the optimization of multiple objectives, the problem model and the solution algorithm have to be modified. We do not consider HRs other than surgeons and nurses. Other HRs involved in running the operating theater and the surgery exist (*e.g.*, anesthesiologists, post-anesthesia care unit HRs, etc.). This work is limited to surgeons and nurses because other HRs that can be relevant (*e.g.*, anesthesiologists) can be modeled in the same way we modeled surgeons and nurses through constraints. HRs like those of the post-anesthesia care unit are not considered as we assume that the post-anesthesia care unit does not limit surgery in any way. In the remainder of this section, we introduce the relevant literature and provide an exhaustive positioning of the paper.

Surgery scheduling is usually treated as an operational problem with a planning horizon spanning one or several days. As pointed out in [4], surgery scheduling involves two main decisions: (i) the assignment of surgical cases to OR sessions (*advanced scheduling*) and (ii) surgical case sequencing for ORs (*allocation scheduling*). Some papers focus only on advanced scheduling (*e.g.*, [16] and [10]) or allocation scheduling (*e.g.*, [3] and [32]) because, at the cost of suboptimal solutions, the two decisions can be taken sequentially in different decision epochs. Relevant papers dealing with both the advanced and allocation scheduling solved sequentially are [20] and [11]. Some recent works focus on advanced and allocation scheduling as an integrated problem to overcome the suboptimality of sequential decisions. In these works, the problem is generally presented as IORPS; relevant papers are [23, 24, 5] and [17]. A multi-operating theater (OT) variant of the IORPS problem is proposed in [29].

In [23], a variant of the general IORPS problem considering surgery waiting lists with up to 1000 surgeries and 6 ORs is solved by combining an integer programming (IP) solver and an improvement heuristic. The problem spans a planning horizon of one week and includes sequence constraints and capacity constraints for ORs and surgeons, due date constraints for surgeries, and some practical constraints for the starting time of surgeries and the

assignment of surgical specialties to ORs. A higher OR utilization rate is targeted by the problem objective, and the obtained solutions increase the OR utilization rate up to 40% with respect to hospital planning. The quality of solutions delivered by the IP solver is good, and the improvement heuristic further improves solutions by a few percentage points; however, the IP solver computation time is large in most cases (on the order of hours). While the results are promising, there is room for improvement in the solution approach. To solve the same problem proposed in [23], the same authors have developed in [24] a structured genetic algorithm able to quickly improve the best solution obtained in the previous work, but the quality of the improvement is in a limited range (few percentage points).

The instances used in [23] and [24] are also used in [28] to test the performances of a branch and check algorithm that combines constraint programming and integer linear programming. The solution approach proposed in [28] outperforms the commercial solver running a pure integer programming model and appears competitive against the approach combining integer programming and the genetic algorithm of [24].

In [5], the same IORPS problem of [23, 24] is decomposed in advanced (master) and allocation (sub) problems. An IP solver-based algorithm relying on generalized disjunctive programming (GDP) modeling of master and subproblems is applied for the problem solution. The mixed-integer linear programming (MILP) convex hull reformulation of the GDP model of both the advanced and allocation problem is given, and a hybrid continuous-time formulation with multiple time-grid (one per OR) and general precedence sequencing variables is adopted for the allocation problem. The authors promote the hybrid formulation (see [6]) as the best solution for the allocation problem. The obtained solutions quality is good, but the computation time is large (more than an hour).

An advanced branch-and-price-and-cut (B&P&C) algorithm is proposed in [17] for a IORPS problem considering sequence constraints for infectious/noninfectious surgeries and capacity constraints for surgeons. The CG algorithm is based on the time-indexed formulation of the OR scheduling problem proposed in [23, 24], the CG decomposition and the master problem formulation are inspired by [10], but subproblems are modeled by means of constraint programming and consider fine details of OR schedules. Dominance

rules for columns and an infeasibility-detection algorithm are developed to speed up the column generation, and the cutting procedure is based on lifted minimal cover inequality constraints (see [15] and [1]). Numerical results show the efficacy of the dominance rules and of the infeasibility-detection algorithm. The B&P&C algorithm outperforms the benchmark commercial solver, but the computation time is very large (on the order of hours) for tested medium-sized instances (6 ORs and up to 120 surgeries).

A logic-based Benders' decomposition (LBB) (see [18]) is applied in [29] to solve a multi-OT IORPS problem integrated with the surgeon-to-surgery assignment. The available OR capacity is spread over some hospitals, and surgeries are assigned to surgeons according to patients' preferences. The decomposition master problem integrates advanced scheduling and surgeon assignment, and the subproblems are the OR allocation scheduling tasks. Numerical experiments reveal that Benders' decomposition methods are promising for IORPS problems and that computation times are not excessively large for medium-sized instances. The same solution approach of [29] is applied to a single-hospital IORPS problem, the paper focuses on the economic advantages of optimized solution.

In [28], the LBB approach of [29] is adapted to solve the instances of [17]. The authors compare the LBB and a branch-and-check algorithm (very close to the LBB but using constraint programming) against the B&P&C of [17] and the commercial solver running three different models: a pure MILP model, a pure constraint programming model, and one mixing constraint programming and MILP. Such a computational study did not reveal one of the models as strictly superior. The pure constraint programming model provides the smaller gap on average, but the pure MILP solves to optimality more instances. Independently of the run model, the commercial solver seems to be, by far, less effective than the *ad-hoc* algorithms as LBB and B&P&C. Compared to the B&P&C of [17], the LBB provides smaller optimality gaps for some instances, but larger for some others. The B&P&C seems to be more effective for large instances with several ORs and the LBB more competitive for instances with few ORs and short duration surgeries. The two solution approach appears of comparable quality.

Nevertheless, few papers consider HR other than

surgeons. In [27], the authors address an IORPS problem with HR constraints, in which the modeling is inspired by resource-constrained project scheduling and a time unit of 15 minutes for the time discretization is adopted; the MILP formulation is effective only for very small-sized instances, and a heuristic algorithm is proposed for solving relatively large-sized instances (7 ORs and up to 80 surgeries). In [30], two IP-based heuristic methods are proposed to solve an IORPS problem involving HR constraints within restrained computation times at the expense of lower quality solutions; the time is discretized in very large units (30 minutes) to tackle large-sized instances with up to 11 ORs but no more than 90 surgeries.

The IORPS problem addressed in this paper includes constraints that are the most common in the literature for this kind of problem and provides an attempt for synthesizing state-of-the-art modeling of this kind of problem. Surgery due dates are hard constraints, as in [27, 23] and [17], as are readiness dates, as in [27].

In the instances that we consider, only a part of the surgeries has the due date in the time horizon of the problem. So, it is unlikely that the problem is infeasible because of the due date constraints. As a last resort, if the problem is infeasible, one or more surgeries with a strict due date can be removed to obtain a feasible problem and executed in the emergency ORs of the operating theater [22]; but this is a matter that is up to the clinical evaluation of surgeons.

The surgeon availability is given as in [27, 23] and [17], and the OT upstream and downstream resources do not bind the scheduling of surgeries, as in [27, 23] and [17]. The scheduling of surgeries is instead constrained by the availability of specialized HRs other than surgeons, *i.e.*, nurses, as in [27] and [30]. As in [17], OR cleaning times depending on the sequence of surgery infection types are enforced and a very small unit for the time discretization is adopted. We assume that an OR cannot be shared by several specialties on the same day as in [23, 24], a practical constraint not well covered by the existing literature.

As argued in [17], time-indexed models, such as those of [23] and [27], are frequently used. Continuous time models involving big-M constraints, *e.g.*, [33], usually provide weaker linear programming (LP) relaxation and slow down the optimization time. The time-indexed formulation is also used in this paper.

From a practical point of view, failures to account for the most important features of the IORPS with HR constraints constitute the main obstacle to the implementation of surgery planning/scheduling algorithms in many hospitals. Such a problem is hard to solve, even in its simplified version without HR constraints. Simple scheduling algorithms usually fail to provide good (or even just feasible) solutions and lead to a reduced quality of service (*e.g.*, unmatched patient due dates) and direct or indirect economic losses (unused capacity, unscheduled surgeries, etc.). Weak (incomplete) scheduling (done manually, for example) frequently produces infeasible schedules because not all relevant details are considered and material or human resource clashes of surgeries can arise.

The remainder of this paper is organized as follows. In Section 2, the IORPS with HR constraints problem is described. The master problem formulation of the column generation decomposition is presented in Section 3 and that of one of the pricing problems in Section 6. The label-correcting algorithm we develop for solving the pricing problems is described in Section 6.2, and the cutting procedure we develop to improve the master problem LP relaxation is elaborated in Section 4. The B&P algorithm and the branching rules are described in Section 5. The computational experiments through which we test the effectiveness of the developed algorithms are described in Section 7 with the report of the numerical results. The label-correcting algorithm, the Benders' cutting procedure and the evidence of their effectiveness, even for real-world-sized instances, represent the main contributions of this paper.

2. Problem Statement

In this work, we address the problem of surgery scheduling for an OT composed of several ORs and a planning horizon of several days. The scheduling of ORs is constrained by the availability of two types of skilled HRs required for the execution of surgeries, *i.e.*, surgeons and nurses. The surgeon availability is characterized by a maximum daily time for surgery, whereas nurses are available according to a calendar. Every surgeon and every nurse belongs to a surgical group (SG), and an SG is a collection of surgical specialties characterized by surgical similarities (*e.g.*, head surgery, chest surgery, etc.). The surgeries to schedule are selected

from a wait list. Each surgery is characterized by the surgical specialty, the duration, the infection type, the number of nurses required, the revenue, the release date and the due date. Surgeries with different infection types were first studied in [3], and then in [17], information on surgery infection types was required for correct scheduling of OR cleaning times. Revenue is a financial measure, but without changing the problem nature, it can be easily replaced by another measure such as a score calculated according to a set of criteria (*e.g.*, priority) or simply the surgery duration as in [17]. Surgeries with the due dates falling in the planning horizon are mandatory surgeries; these are surgeries that must be scheduled in the planning horizon, whereas other surgeries may be scheduled or not. The following assumptions hold throughout the paper.

1. The duration of each surgery is deterministic.
2. Every surgery is already assigned to a surgeon.
3. The maximum opening time of the ORs is eight hours and there is no OR or nurse overtime.
4. There is no fixed OR opening cost.
5. The ORs composing the OT are identical with respect to the surgery execution.
6. OT upstream and downstream resources do not constrain the OR scheduling.
7. Nurses can only assist in the execution of surgeries of their own SG.
8. Nurses are trained to assist the execution of surgeries as either scrub or circulating nurses.
9. An obligatory cleaning time (OCT) of the OR may be required between two surgeries of a different infection type, where the OCT depends exclusively on the immediately preceding surgery. Moreover, we assume that:
 - (a) a surgery infection type is characterized by an infection level,
 - (b) infection types can be sorted according to the infection level,
 - (c) OCTs are required to switch from infection types of higher infection levels to those of lower infection levels and
 - (d) given any pair of infection types with infection levels f and f' such that $f > f'$ and $OCT(f, f')$ being the OCT required to switch from infection level f to f' , it holds that $OCT(f, f') \leq \sum_{g=f, \dots, f'-1} OCT(g, g+1)$. This is a special case of the triangular inequality, *i.e.*, it is more efficient (faster) to downgrade

the OR infection level from f to f' directly with a unique OCT than by passing through the intermediate steps with a sequence of OCTs.

As in many works cited in Section 1, *e.g.*, [17, 23, 24, 6], Assumption 1 holds as the problem is already very hard to solve in its deterministic formulation; exact methods are able to solve with sufficient solution accuracy only very small instances of IORPS problems in the case of uncertain duration of surgeries [2]. Assumption 2 reflects the common practice of many hospitals. Assumption 2 is consistent with the practice of a wide range of hospitals, from university hospitals where each surgeon has her/his study cases to private hospitals where freelance surgeons perform surgeries on their patients. Eight hours is the regular open time for elective surgeries in many hospitals, and marginal costs of surgery, such as overtime or OR openings, are considered in some papers. However, we do not consider marginal costs, *i.e.*, Assumption 3 and Assumption 4, for the following reasons: (1) OR or nurse overtime is an undesirable event that may occur in the OT but should be avoided as much as possible; it thus seems unreasonable to schedule some overtime in advance as a regular practice, and (2) the marginal cost of allowing an OR to remain open is mainly given by its staffing. If the surgical staff is given (*i.e.*, the cost for staffing is already incurred), the remaining marginal cost is very low and negligible and given by surgical consumables. In regard to Assumption 5, only a few papers consider the opposite case of different OR equipment, and this does not provide a significant contribution. Assumption 6 is realistic for many hospitals. Assumption 7 is very realistic because nurses are usually assigned to an SG in the long term and are trained to assist only surgeries of their SG, whereas only veteran nurses have sufficient experience to assist, within a short time horizon (the same day or week), the execution of surgeries belonging to different SGs. Assumption 8 follows the common practice of many hospitals. Assumption 9 is as in [17] and is very realistic and necessary, but done in isolation as presented in [17], it is not sufficient for achieving full consistency with real-world functioning of ORs and OCT scheduling.

We extend Assumption 9 with Assumptions 9a, 9b, 9c and 9d; otherwise, by neglecting Assumptions 9a-9d, it leads to surgeries with “cleaning power”, which is a paradox. To better clarify, con-

sider the schedule of three surgeries i_1, i_2 and i_3 in the same OR with $OCT(i_1, i_2) = OCT(i_2, i_3) = 0$ but $OCT(i_1, i_3) > 0$ which does not meet Assumptions 9c. The sequence $i_1 \prec i_2 \prec i_3$ contains no OCT at all, and the insertion of i_2 between i_1 and i_3 has necessarily the paradoxical “cleaning power” on the OR.

The problem objective is the maximization of the total revenue of scheduled surgeries. A solution of the problem is obtained with the following decisions: (1) to assign SGs to ORs on a daily basis, (2) to select the surgeries to schedule, (3) to assign every selected surgery to an OR and (4) to assign a starting time to every selected surgery. These decisions are all correlated with respect to the problem objective, and an integrated decision is required to avoid suboptimal solutions. A feasible problem solution must respect the constraints that follow. Mandatory surgery must be scheduled only once in the planning horizon, and other surgeries may be scheduled at most once. A surgery can be scheduled for any day between its release date and due date, but not before/after its release/due date. The total daily OR opening time and the daily maximum surgery time of surgeons cannot be exceeded. The required number of nurses have to be available for the execution of each surgery. Surgeries cannot overlap in the same OR, and surgeries of the same surgeon cannot overlap. For every OR and day, only surgeries of the assigned SG can be scheduled in the OR. OR cleaning must be scheduled according to the OR sequence of surgeries and their infection type. No cleaning is required between two cases with no infection or the same type of infection. In Table 1, we report the notations for the problem we describe.

The MILP of the problem, given in Appendix, is essentially the same as that of many other works on similar subjects, *e.g.*, [17, 23, 24], but with additional constraints for HRs other than surgeons (*i.e.*, nurses).

3. Column Generation Master Problem

In this section, we present the master problem (MP) of the column generation reformulation of the IORPS with HR constraints problem. In such an MP, a variable (*i.e.*, a column) stands for a schedule of one OR and one day in the planning horizon.

Let Γ^K be the set of all OR schedules and subset $\Gamma_j^K \subset \Gamma^K$ be the set of feasible schedules for day $j \in J$. Let r_γ^i be a binary value that takes value 1 if surgery i belongs to schedule $\gamma \in \Gamma^K$ and r_γ^{at} be a binary value that takes value 1 if surgeon a is occupied with surgery at time t in schedule $\gamma \in \Gamma^K$. Let also C_γ be the revenue of schedule $\gamma \in \Gamma^K$; this is the sum of the revenues of surgeries belonging to the schedule. Value p_γ^a is the total surgery time for surgeon a in schedule $\gamma \in \Gamma^K$, and value n_γ^{mt} is the number of nurses of SG m occupied with surgery at time t in schedule $\gamma \in \Gamma^K$. The decision variable ϑ_γ takes value 1 if the schedule $\gamma \in \Gamma_j^K$ is selected for day j (0, otherwise). The OR master problem (ORMP) formulation reads:

$$\max \sum_{\gamma \in \Gamma^K} C_\gamma \vartheta_\gamma \quad (1)$$

subject to

$$\sum_{j \in J} \sum_{\gamma \in \Gamma_j^K} r_\gamma^i \vartheta_\gamma = 1 \quad (i \in I_{dd}), \quad (2)$$

$$\sum_{j \in J} \sum_{\gamma \in \Gamma_j^K} r_\gamma^i \vartheta_\gamma \leq 1 \quad (i \in I \setminus I_{dd}), \quad (3)$$

$$\sum_{\gamma \in \Gamma_j^K} r_\gamma^{at} \vartheta_\gamma \leq 1 \quad (a \in A; j \in J; t \in T), \quad (4)$$

$$\sum_{\gamma \in \Gamma_j^K} p_\gamma^a \vartheta_\gamma \leq T_{aj} \quad (a \in A; j \in J), \quad (5)$$

$$\sum_{\gamma \in \Gamma_j^K} n_\gamma^{mt} \vartheta_\gamma \leq |B_{mj}| \quad (m \in SG; j \in J; t \in T), \quad (6)$$

$$\sum_{\gamma \in \Gamma_j^K} \vartheta_\gamma \leq |K_j| \quad (j \in J), \quad (7)$$

$$\vartheta_\gamma \in \{0, 1\} : \gamma \in \Gamma^K, \quad (8)$$

The objective function (1) maximizes the total surgery revenue for surgeries scheduled in the planning horizon. Constraints (2) and (3) enforce that mandatory surgeries are scheduled once in the planning horizon and nonmandatory surgeries are scheduled at most once in the planning horizon. Constraints (4) enforce that surgeries of the same surgeon do not overlap: a surgeon can execute only one surgery at a time. Constraints (5) enforce that

Notation	
<i>Sets</i>	
A	Surgeons
B	Nurses
I	Surgeries
J	Days in the planning horizon
K	Operating rooms
SG	Surgical groups
T	Time slots in a day
F	OR cleaning times
<i>Subsets</i>	
B_m	Nurses of SG $m \in SG$
B_{mj}	Nurses of SG $m \in SG$ available on day $j \in J$
I_{dd}	Mandatory surgeries
I_a	Surgeries of surgeon $a \in A$
I_m	Surgeries of SG $m \in SG$
I_j	Surgeries that can be scheduled for day $j \in J$
I_{mj}	Surgeries of SG $m \in SG$ that can be scheduled for day $j \in J$
K_j	Operating rooms open on day $j \in J$
<i>Parameters</i>	
T_{aj}	Maximum surgery time of surgeon $a \in A$ for day $j \in J$
c_i	Revenue for surgery $i \in I$
rd_i	Release date of surgery $i \in I$
dd_i	Due date of surgery $i \in I$
p_i	Duration of surgery $i \in I$
$OCT(i, i')$	Duration of the OCT to execute after surgery i and before surgery i'
	The equivalent of $OCT(f_i, f_{i'})$
n_i	Required number of nurses to assist the execution of surgery $i \in I$
f_i	The infection level of surgery i

Table 1: Problem notation

the maximum daily time of surgery is not exceeded for every surgeon and day in the planning horizon. Constraints (6) enforce that, for every SG, day and time, the number of nurses occupied with surgery never exceeds the number of available nurses for the given SG and day. Constraints (7) enforce that at most $|K_j|$ OR schedules are selected for every day in the planning horizon. The domain of variables ϑ is defined by (8).

The LP relaxation of formulation (1)-(8) is solved by means of column generation techniques given the large, potentially huge, number of variables (feasible schedules) $\gamma \in \Gamma^K$. Column generation techniques tackle such huge numbers of variables considering a restricted number of variables, the subset $\tilde{\Gamma}^K \subset \Gamma^K$. The MP including only the restricted

set of variables $\tilde{\Gamma}^K$ is the restricted master problem (RMP). Let ORRMP be the OR restricted MP.

The master problem of the column generation scheme presented in this section is essentially the same as that of [17]; we added the resource constraints (6) for the nurse availabilities. In [17], the formulation of the master problem is solved by means of column generation (as in our work) and by resorting to a constraint programming (CP) model for solving subproblems. Since computational efficiency is crucial for generating new columns, we preferred to devise a label-correcting algorithm for solving subproblems, see Section 6.2, because we think that such an approach is likely faster than CP for generating new columns. It is also known that CP suffers from significant overhead times for

the model solutions, as in integer linear programming (ILP), and thus, it may not be the best for the generation of columns.

4. Benders' Cutting Procedure

In this section, we describe the Benders' cutting procedure that we developed for tightening the LP relaxation of model (1)-(8). This cutting procedure is inspired by the cutting phase of the three-phase algorithm proposed in [7] for the simultaneous assignment of locomotives and cars to passenger trains. In [7], the classic Benders' decomposition is combined with column generation to decompose and efficiently solve the problem. The same three-phase algorithm is successfully applied for the simultaneous scheduling of aircraft and crew as well; see [8, 25, 26]. Both the simultaneous assignment of locomotives and cars to passenger trains and the simultaneous scheduling of aircraft and crew present a formulation that uses a type of column variable for each resource. In both cases, once Benders' decomposition is applied, the master problem and the subproblem have column variables of/for, respectively, only one type/resource, and the applied Benders' decomposition is also a decomposition by resources.

The cutting procedure that we developed is inspired by Benders' decomposition and developed around the idea of a master resource and some slave resources. Each resource type has its own problem, and the solution of the master problem (that of the master resource) is enforced in the subproblems (those of slave resources) for generating cuts. ORs are the master resource, and surgeons and nurses are slave resources. Our approach differs from a classic Benders' decomposition because the OR solution that we enforce is partial. For generating cuts, we determine computationally efficient to enforce in subproblems the sole surgery selection given a complete schedule for the ORs; this is at the price of retaining resource constraints for surgeons and nurses in the master problem formulation. Through the generation of cuts and the convergence of the master problem and subproblems on a common selection of surgery and objective function value, the LP relaxation of the problem is tightened.

The ORMP of Section 3 is the master problem of the cutting procedure and the two Benders' subproblems are the *surgeon subproblem* (SSP) and the *nurse subproblem* (NSP). The SSP schedules surg-

eries for each surgeon and day in planning horizon. The NSP schedules surgeries for each nurse and day in planning horizon. At each iteration, for both the SSP and the NSP, the selection of surgeries is constrained by the current LP solution $\bar{\vartheta}_\gamma : \gamma \in \Gamma^K$ of the ORMP. Each time a Benders' subproblem is solved, a new cut can be added to the ORMP. The Benders' cutting procedure stops when the objective function of both the SSP and the NSP equal that of the ORMP, *i.e.*, the objective functions of the three problems have converged on the same value. In Figure 1, we present the flow diagram of the Benders' cutting procedure in relation to the basic column generation algorithm.

In Section 4.1, we report the formulation of the *surgeon subproblem* (SSP) and, in Section 4.2, that of the *nurse subproblem* (NSP). In Section 4.3, we report the restatement of the ORMP with included the formulation of the Benders' cuts generated through the SSP and NSP.

4.1. Surgeon subproblem

Let Γ_j^a be the subset of feasible schedules of surgeon $a \in A$ for day $j \in J$ and Γ^A be the set of all feasible schedules of all surgeons A . Let also C_γ be the total revenue of schedule $\gamma \in \Gamma^A$ defined as the sum of the revenues of surgeries belonging to the schedule. Decision variable ζ_γ defines whether the schedule $\gamma \in \Gamma^A$ is selected or not. Given nonnegative values $\bar{\vartheta}_\gamma$ such that $\gamma \in \Gamma^K$ satisfying constraints (2)-(7), the SSP reads:

$$\max \sum_{\gamma \in \Gamma^A} C_\gamma \zeta_\gamma \quad (9)$$

subject to

$$\sum_{j \in J} \sum_{\gamma \in \Gamma_j^a} r_\gamma^i \zeta_\gamma \leq \sum_{\gamma \in \Gamma^K} r_\gamma^i \bar{\vartheta}_\gamma \quad (a \in A, i \in I_a), \quad (10)$$

$$\sum_{\gamma \in \Gamma_j^a} \zeta_\gamma \leq 1 \quad (a \in A; j \in J), \quad (11)$$

$$\zeta_\gamma \geq 0 : \gamma \in \Gamma^A. \quad (12)$$

A solution of the SSP defines a schedule of surgery for every surgeon and day in the planning horizon. The SSP objective (9) maximizes the total revenue of the surgeries scheduled in the planning horizon. Constraints (10) bound the selection of each surgery

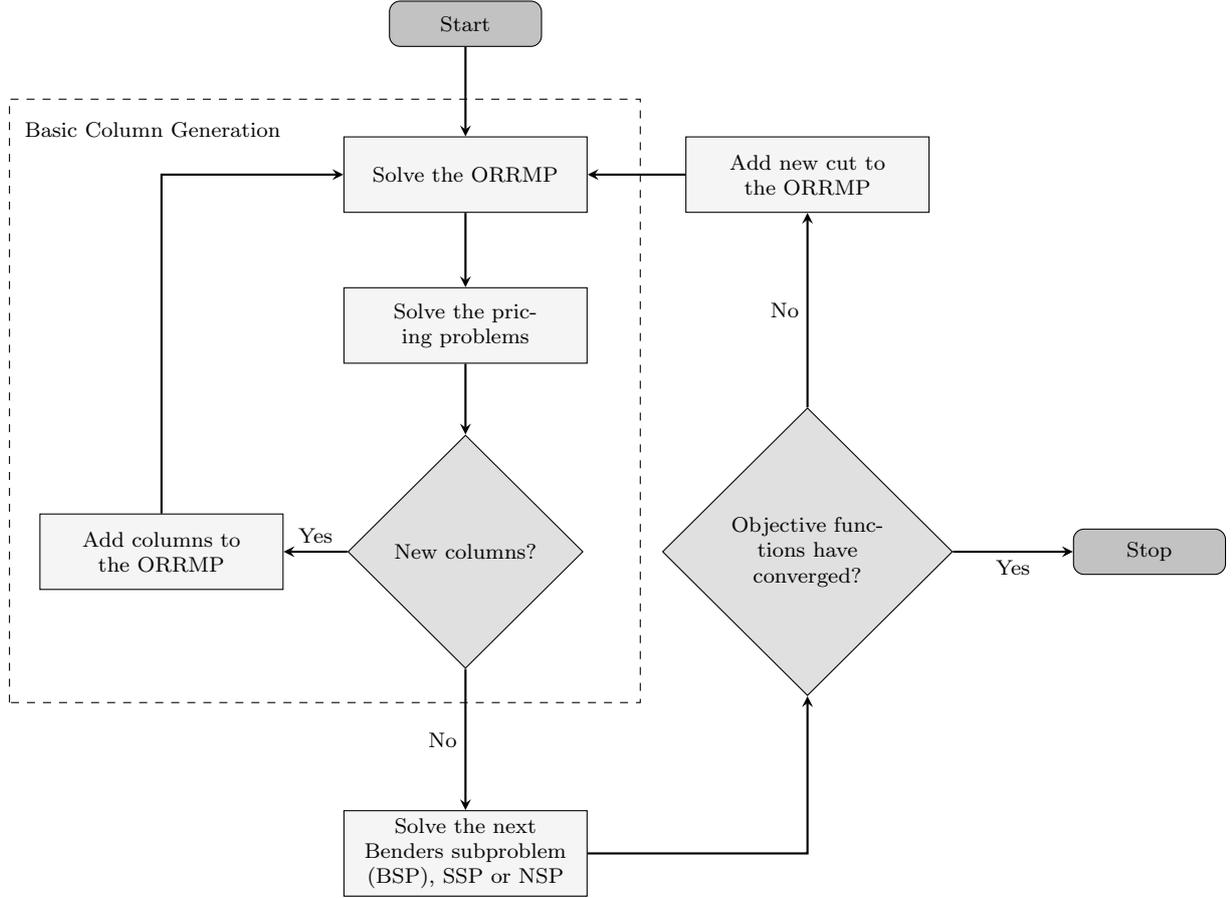


Figure 1: Column generation with Benders cutting procedure

in the range $[0; \sum_{\gamma \in \Gamma^K} r_\gamma^i \bar{\vartheta}_\gamma]$. Constraints (11) enforce that at most one schedule is selected for each day and each surgeon.

Note that to enforce that mandatory surgeries must be scheduled is not necessary for the SSP and constraints on mandatory surgeries are thus relaxed. To enforce the equality in Constraints (10) for mandatory surgeries is not necessary because the selection of surgeries in the SSP is the same as the ORMP when the two objective functions have converged on the same value with the Benders' cutting procedure. Given $\bar{\vartheta}_\gamma : \gamma \in \Gamma^K$ a solution of (2)-(7) such that the value of the SSP objective function equals that of the ORMP, it holds that

$$\sum_{j \in J} \sum_{\gamma \in \Gamma_j^a} r_\gamma^i \zeta_\gamma = \sum_{\gamma \in \Gamma^K} r_\gamma^i \bar{\vartheta}_\gamma \quad (a \in A, i \in I_a)$$

Let $\lambda^{(10)}$ and $\lambda^{(11)}$ be the dual variables for constraints (10) and (11), respectively. The *dual problem*

of the SSP reads:

$$\min \sum_{i \in I} \left(\sum_{\gamma \in \Gamma^K} r_\gamma^i \bar{\vartheta}_\gamma \right) \lambda_i^{(10)} + \sum_{a,j} \lambda_{a,j}^{(11)} \quad (13)$$

subject to

$$\sum_i r_\gamma^i \lambda_i^{(10)} + \lambda_{a,j}^{(11)} \geq C_\gamma \quad (\gamma \in \Gamma_j^a, j \in J, a \in A), \quad (14)$$

$$\lambda_i^{(10)} \geq 0 : i \in I, \quad \lambda_{a,j}^{(11)} \geq 0 \quad \forall a, j \quad (15)$$

The formulation of the Benders cut derived from the SSP dual problem is reported in Section 4.3.

4.2. Nurse subproblem

Let Γ_j^{Bm} be the set of nurse schedules feasible for day $j \in J$ and surgical group $m \in SG$. Let Γ^B be

the set of all nurse schedules. Decision variable $\varphi_\gamma : \gamma \in \Gamma_j^{B_m}$ defines the number of nurses of surgical group $m \in SG$ with schedule $\gamma \in \Gamma_j^{B_m}$ for day $j \in J$ in the problem solution. Let $C_\gamma : \gamma \in \Gamma^B$ be the schedule total revenue defined as: $\sum_{i \in \gamma} c_i/n_i$. For given nonnegative values $\bar{\vartheta}_\gamma : \gamma \in \Gamma^K$ satisfying Constraints (2)-(7), the NSP reads:

$$\max \sum_{\gamma \in \Gamma^B} C_\gamma \varphi_\gamma \quad (16)$$

subject to

$$\sum_{\gamma \in \Gamma_j^{B_m}} r_\gamma^i \varphi_\gamma \leq n_i \sum_{\gamma \in \Gamma_j^K} r_\gamma^i \bar{\vartheta}_\gamma \quad (i \in I_m, m \in SG), \quad (17)$$

$$\sum_{\gamma \in \Gamma_j^{B_m}} \varphi_\gamma \leq |B_{mj}| \quad (m \in SG; j \in J), \quad (18)$$

$$\varphi_\gamma \geq 0 : \gamma \in \Gamma^B. \quad (19)$$

A solution of the NSP defines a nurse schedule selection for every SG and day in the planning horizon. The NSP objective (16) maximizes the total revenue of the scheduled surgeries. Constraints (17) enforce that if a surgery is scheduled for the ORMP, the surgery can be scheduled at most for the required number of nurses of its SG. Constraints (18) enforce that the number of times that a schedule is selected cannot be greater than the number of available nurses of the related SG. Constraints on mandatory surgeries are relaxed also in the NSP, this is because when the Benders' cutting procedure stops, it holds that

$$\sum_{\gamma \in \Gamma_j^{B_m}} r_\gamma^i \varphi_\gamma = n_i \sum_{\gamma \in \Gamma_j^K} r_\gamma^i \bar{\vartheta}_\gamma \quad (m \in SG, i \in I_m).$$

Let $\lambda^{(17)}$ and $\lambda^{(18)}$ be the dual variables for Constraints (17) and (18), respectively. The *dual problem* of the NSP reads:

$$\min \sum_{i \in I_m} n_i \left(\sum_{\gamma \in \Gamma^K} r_\gamma^i \bar{\vartheta}_\gamma \right) \lambda_i^{(17)} + \sum_{m \in SG} \sum_{j \in J} |B_{mj}| \cdot \lambda_{mj}^{(18)} \quad (20)$$

subject to

$$\sum_{i \in I_m} r_\gamma^i \lambda_i^{(17)} + \lambda_{mj}^{(18)} \geq C_\gamma \quad (m \in SG, j \in J, \gamma \in \Gamma_j^{B_m}), \quad (21)$$

$$\lambda_i^{(17)} \geq 0 \quad \forall i, \quad \lambda_{mj}^{(18)} \geq 0 \quad \forall m, j \quad (22)$$

The formulation of the Benders' cut derived from the NSP dual problem is reported in Section 4.3.

4.3. ORMP restatement and cutting procedure

From dual problems (13)-(15) and (20)-(22), two types of Benders' cuts to add to the ORMP are derived.

Let Δ denote the polyhedron defined by Constraints (14) and (15) of the SSP; let also P_Δ and R_Δ be the set of extreme points and extreme rays of Δ , respectively. Note that, because of relaxed Constraints (10) on mandatory surgeries, the SSP is always feasible, and the set R_Δ of extreme rays is therefore empty. Let Ξ denote the polyhedron defined by Constraints (21) and (22) of the NSP; let also P_Ξ and R_Ξ be the set of extreme points and extreme rays of the Ξ , respectively. The NSP is always feasible because of relaxed Constraints (17) on mandatory surgeries, and the set R_Ξ of extreme rays is therefore empty.

Let $z_0 \in \mathbb{R}^+$ be an auxiliary real variable to express the total revenue of scheduled surgeries. The ORMP, *i.e.*, formulation (1)-(8), can be restated for the Benders' cutting procedure as:

$$\max z_0 \quad (23)$$

subject to

$$(2), (3), (4), (5), (6), (7)$$

$$\sum_i \sum_{\gamma \in \Gamma^K} c_i r_\gamma^i \vartheta_\gamma - z_0 = 0 \quad (24)$$

$$\sum_i \sum_{\gamma \in \Gamma^K} \lambda_i^{(10)} r_\gamma^i \vartheta_\gamma - z_0 \geq - \sum_{a,j} \lambda_{aj}^{(11)} \quad ((\lambda^{(10)}, \lambda^{(11)}) \in P_\Delta), \quad (25)$$

$$\sum_i \sum_{\gamma \in \Gamma^K} n_i \lambda_i^{(17)} r_\gamma^i \vartheta_\gamma - z_0 \geq - \sum_{m,j} |B_{mj}| \cdot \lambda_{mj}^{(18)} \quad ((\lambda^{(17)}, \lambda^{(18)}) \in P_\Xi), \quad (26)$$

$$z_0 \geq 0, \quad \vartheta_\gamma \in [0; 1] : \gamma \in \Gamma^K. \quad (27)$$

Constraint (24) limits the z_0 variable to total revenue of scheduled surgeries expressed as a function of variables $\theta_\gamma : \gamma \in \Gamma^K$; this constraint allows to start the Benders' cutting procedure with an already good LP relaxation of the ORMP. Constraints (25) and (26) are the optimality Benders' cuts generated by solving the SSP and the NSP, respectively.

The optimality Benders cuts, Constraints (25) and (26), are not generated exhaustively. The Benders' cutting procedure generates only a subset of cuts sufficient to obtain an optimal solution. Feasibility Benders' cuts are never generated at all given the emptiness of R_Δ and R_Ξ ; we therefore omit the formulation of feasibility cuts. At each iteration of the algorithm, the relaxed ORMP is solved including the subset of cuts generated until the current iteration, *i.e.*, $P'_\Delta \subseteq P_\Delta$ and $P'_\Xi \subseteq P_\Xi$.

5. Branch and Price

To obtain an optimal, or at least high-quality, integer solution of the ORMP, we developed a branch-and-price (B&P) procedure. Given $\vartheta_\gamma \geq 0 : \gamma \in \Gamma^K$ the current LP solution of the ORMP, the B&P branching is based on the following rules:

1. branch on the total number of surgeries scheduled per SG in the planning horizon;
2. branch on whether a surgery is scheduled for a given day and time or not.

Branching rules 1 and 2 are applied hierarchically and according to the numerical order; rule 2 is applied if, with rule 1, the branching cannot be further applied.

To apply the branching rule 1, given $m' \in SG$, the surgical group with the most fractional number of surgeries scheduled in the current ORMP solution (*i.e.*, with the fractional part of the number closest to 0.5), one of the following two constraints has to be added to the ORMP:

$$\sum_{i \in I_{m'}} \sum_{\gamma \in \Gamma^K} r_\gamma^i \vartheta_\gamma \leq \left\lfloor \sum_{i \in I_{m'}} \sum_{\gamma \in \Gamma^K} r_\gamma^i \bar{\vartheta}_\gamma \right\rfloor, \quad (28)$$

$$\sum_{i \in I_{m'}} \sum_{\gamma \in \Gamma^K} r_\gamma^i \vartheta_\gamma \geq \left\lceil \sum_{i \in I_{m'}} \sum_{\gamma \in \Gamma^K} r_\gamma^i \bar{\vartheta}_\gamma \right\rceil. \quad (29)$$

Let Ω_1 be the set of all possible cuts of form (28) or (29) that can be generated.

To apply the branching rule 2, we branch on the most fractional scheduled time of a surgery in the current solution, *i.e.*, given r_γ^{ijt} a binary value that takes value 1 if surgery i is scheduled for day j and time t in schedule γ , one of the following two constraints has to be added to the ORMP:

$$\sum_{\gamma \in \Gamma^K} r_\gamma^{ijt} \vartheta_\gamma = \left\lfloor \sum_{\gamma \in \Gamma^K} r_\gamma^{ijt} \bar{\vartheta}_\gamma \right\rfloor, \quad (30)$$

$$\sum_{\gamma \in \Gamma^K} r_\gamma^{ijt} \vartheta_\gamma = \left\lceil \sum_{\gamma \in \Gamma^K} r_\gamma^{ijt} \bar{\vartheta}_\gamma \right\rceil. \quad (31)$$

Note that $0 \leq \sum_{\gamma \in \Gamma^K} r_\gamma^{ijt} \bar{\vartheta}_\gamma \leq 1$ for every $t \in T_j$, $j \in J$ and $i \in I$. Let Ψ be the set of all possible constraints of this type that can be generated.

Constraints in Ω_1 are effective for improving the column generation upper bound and detecting the problem infeasibility of a given B&P node, and constraints Ψ drive the search of an integer solution for the restricted ORMP. An integer solution for each node of the B&P is computed by enforcing integrality constraints to the column variables added to the ORMP and by solving the ORMP. The best integer solution of the ORMP found until the current iteration of the B&P provides a lower bound on the optimal solution of the problem and allows an early pruning of B&P nodes.

6. Column Generation Subproblem

The three problem formulations presented in Section 4 imply column generation for their solution. In this section, we describe the solution method that is applied for solving the pricing problems of the ORMP. The pricing problems of the SSP and the NSP are simplified versions of ORMP ones. So, we describe the solution method of the OR pricing problem (ORPP) and, when necessary, we point out what changes hold for solving the surgeon pricing problem (SPP) or the nurse pricing problem (NPP).

In Section 6.1, we define the ORPP as an extension of the rainbow path (RP) problem, see [21], that we call the maximum revenue RP with resource constraints (MRRPRC) problem, since searching for the OR schedule with maximum reduced cost is equivalent to searching for the MR-

RPRC over a graph. In Section 6.2, we describe the RP algorithm developed for solving the MRRPRC problem.

6.1. Description of the MRRPRC

Let $G = (V, E)$ be a directed acyclic graph (DAG), where V is the set of nodes and E is the set of arcs. There is a node v_t for each possible starting time $t = 0, \dots, T$ of every surgery and three types of arcs: *surgery arcs*, *idle time arcs* and *OR-cleaning time arcs*. There is a surgery arc (i, v_t) of length p_i that reaches node v_{t+p_i} for every surgery $i \in I$ and feasible starting time t . There is an idle time arc from v_t to v_{t+1} for every time period $t \leq T - p_i$. There is an OR-cleaning time arc from v_t to $v_{t+OCT(f, f')}$ for every pair of infection levels (f, f') such that $OCT(f, f') > 0$ and time period $t \leq T - OCT(f, f')$. Every surgery arc that corresponds to the same surgery has the same color $i \in I$. Idle time and OR-cleaning time arcs have no color.

Let $d_{ai} \geq 0$ be the consumption of resource (surgeon) $a \in A$ along any arcs of color i , it corresponds to the surgery duration (arc length) $d_{ai} = p_i$. For every resource $a \in A$, the capacity Q_a is defined.

Note that, in this MRRPRC problem, the graph is acyclic (any path is elementary by graph construction) and pairs of nodes are linked by many arcs instead of single arcs (as often encountered in column generation).

Let λ be the vector of dual costs of constraints from (2) to (6) of the master problem formulation plus the Benders cuts' constraints, (25) and (26), the branching constraints (30) or (31) and (30) or (31). For each element of λ , the constraint reference is in superscript and the constraint indices are subscripts, e.g., $\lambda_i^{(2,3)}$ are the dual costs of Constraint (2) or (3) depending on whether the surgery $i \in I$ is mandatory or not, respectively. Let x_{it} be a binary decision variable that takes value 1 if arc (i, v_t) is selected (0 otherwise). The reduced cost of

a schedule $\gamma \in \Gamma^K$ is given by

$$\begin{aligned}
C_\gamma &= \sum_{i \in I_j} \sum_{t \in T_j} \lambda_i^{(2,3)} x_{it} - \sum_{i \in I_j} \sum_{t \in T_j} \sum_{t'=t}^{t+p_i-1} \lambda_{a_{ij}t'}^{(4)} x_{it} \\
&\quad - \sum_{i \in I_j} \sum_{t \in T_j} \lambda_{a_{ij}}^{(5)} p_i x_{it} \\
&\quad - \sum_{i \in I_j} \sum_{t \in T_j} \sum_{t'=t}^{t+p_i-1} \lambda_{m_{ijt'}}^{(6)} n_i x_{it} \\
&\quad - \sum_{i \in I_j} \sum_{t \in T_j} \sum_{\nu \in P'_\Delta} \lambda_\nu^{(25)} \lambda_i^{(10)} x_{it} \\
&\quad - \sum_{i \in I_j} \sum_{t \in T_j} \sum_{\nu \in P'_\Xi} \lambda_\nu^{(26)} n_i \lambda_i^{(17)} x_{it} \\
&\quad - \sum_{\omega \in \Omega'_1} \lambda_\omega^{(28,29)} \sum_{i \in I_j} \sum_{t \in T_j} x_{it} \\
&\quad - \sum_{(i,j,t) \in \Psi'} \lambda_{ij}^{(30,31)} x_{it} - \lambda_j^{(7)}
\end{aligned} \tag{32}$$

that is separable according to starting times of surgeries; the part of the reduced cost depending on surgery $i \in I$ and time $t \in T_j$ is given by

$$\begin{aligned}
w_{it} &= c_i - \lambda_i^{(2,3)} - \sum_{t'=t}^{t+p_i-1} \lambda_{a_{ij}t'}^{(4)} - \lambda_{a_{ij}}^{(5)} p_i \\
&\quad - \sum_{t'=t}^{t+p_i-1} \lambda_{m_{ijt'}}^{(6)} n_i - \sum_{\nu \in P'_\Delta} \lambda_\nu^{(25)} \lambda_i^{(10)} \\
&\quad - \sum_{\nu \in P'_\Xi} \lambda_\nu^{(26)} n_i \lambda_i^{(17)} - \sum_{\omega \in \Omega'_1} \lambda_\omega^{(28,29)} \\
&\quad - \sum_{(i',j,t) \in \Psi': i'=i} \lambda_{i'jt}^{(30,31)} x_{i't},
\end{aligned} \tag{33}$$

and the reduced cost can be written as

$$\sum_{i \in I_j} \sum_{t \in T_j} w_{it} x_{it} - \lambda_j^{(7)}. \tag{34}$$

Every surgery arc thus has revenue $w_{(i,v_t)} = w_{it}$. Idle time and OR-cleaning time arcs have revenue equal to zero.

The problem objective is to identify the RP from v_0 to v_T that satisfies resource constraints and maximize the revenue. The MRRPRC can be described with the following MILP:

$$\max \sum_{(i,v_t) \in E} w_{it} x_{it} - \sum_{m \in SG} -\lambda_j^{(7)} \tag{35}$$

subject to

$$\sum_{(i,v_t) \in E} x_{it} - \sum_{(i,v_t) \in E} x_{it} = 0 \quad (v_t \in V \setminus \{v_0, v_T\}), \quad (36)$$

$$\sum_{(i,v_0) \in E} x_{it} = 1, \quad (37)$$

$$\sum_{(i,v_T) \in E} x_{it} = 1, \quad (38)$$

$$s_{at} + d_{ai} \leq s_{ait+p_i} + M(1 - x_{it}), \quad (39)$$

$$s_{at} \leq Q_a, \quad (40)$$

$$\sum_{m \in SG} y_m \leq 1, \quad (41)$$

$$\sum_{i \in I} \sum_{(i,v_t) \in E} x_{it} \leq y_m \quad (m \in SG), \quad (42)$$

$$\sum_{(i,v_t) \in E} x_{it} \leq 1 \quad (i \in I), \quad (43)$$

$$\sum_{(i',v_{t'}) \in E_{i,(i')}}^{oct} x_{i't'} \leq 1 - x_{it} \quad ((i, i') \in I : i \neq i', v_t \in V), \quad (44)$$

$$x_{it} \in \{0; 1\} \quad \forall i \in I_j, t \in T_j. \quad (45)$$

Binary decision variable y_m takes value 1 if arcs of colors belonging to SG m can be selected (0 otherwise). Variable s_{at} measures the consumption of resource a reaching node v_t . The objective function (35) maximize the revenue, *i.e.*, search for the best paths. Constraints (36)-(40) are those for resources. Constraints (41) and (42) enforce that only colors of the selected SG can be selected. Constraints (43) enforce that paths have to be rainbow paths. Constraints (44) enforce infection level constraints on the sequence of surgery and OR cleaning arcs. With (45), the domain of variables x_{it} is defined.

The revenue w_{it} of the surgery arcs (i, v_t) have a simpler formulation in the SPP graph and the NPP graph, these are $w_{it} = c_i - \lambda_i^{(10)}$ and $w_{it} = c_i - \lambda_i^{(17)}$, respectively. The objective function becomes

$$\max \sum_{(i,v_t) \in E} w_{it}x_{it} - \lambda_{aj}^{(11)} \quad (46)$$

for the SPP and

$$\max \sum_{(i,v_t) \in E} w_{it}x_{it} - \lambda_{mj}^{(18)} \quad (47)$$

for the NPP.

In both the graph of the SPP and the NPP, there are no OR-cleaning time arcs and path coloring Constraints (41) and (42) have to be removed from the model. Even if there is no term that is a function of the time index t in the formulation of the surgery arc revenue w_{it} , the proposed graph modeling is effective as it allows considering constraints on surgery scheduling, *i.e.*, constraints in Ψ .

In the context of column generation, the MR-RPRC optimization can be terminated as soon as some feasible paths are found, namely, paths that for the ORPP, satisfy

$$\sum_{(i,v_t) \in E} w_{it}x_{it} - \lambda_j^{(7)} > 0, \quad (48)$$

i.e., columns with a strictly positive reduced cost. As any feasible schedule can be represented as a path in the graph, the absence of such a rainbow path means that there exist no variables with a positive reduced cost with the given set of dual variables. A condition similar to (48) can be derived for the SPP and the MPP from objectives (46) and (47), respectively.

Proposition 1. *The MRRPRC problem is strongly NP-Hard.*

PROOF. We prove the theorem by reducing the U problem to the MRRPRC problem. Problem U is a variant of the *single-machine scheduling problem with electricity costs*, and it is proven to be strongly NP-Hard by reduction of the 3-PARTITION problem; see [9]. Problem U is as follows. Jobs \mathcal{J} must be processed nonpreemptively at a uniform speed over a time horizon of distinct and contiguous time periods $\mathcal{P} = \{1, \dots, |\mathcal{K}|\}$. Let a_k and d_k be, respectively, the starting time and the duration of time period $k \in \mathcal{P}$. Each time period $k \in \mathcal{P}$ has an electricity price c_k per unit of energy. Each job $j \in \mathcal{J}$ has processing time p_j and a power demand q_j . It is assumed that processing times and the duration of time periods are given as integers. The relationship between processing time and power demand is arbitrary. We can create an instance of the MRRPRC problem from any instance of the U problem as follows. Let $d_{jkt} = [t; t + p_j - 1] \cap [a_k, a_k + d_k - 1]$

be the processing time of job j that overlaps time period k if the job starts at time t . For each job $j \in \mathcal{J}$, a surgery i with execution time $p_i = p_j$ and revenue $w_{it} = C - \sum_{k \in \mathcal{P}} c_k q_j d_{jkt}$ is created given C a constant value such that $C \geq \sum_{k \in \mathcal{P}} c_k q_j d_{jkt}$ for every $j \in \mathcal{J}$ and $t \in T$.

6.2. Rainbow path algorithm

The idea of modeling the problem as the search of paths over a graph is inspired by [31] where the Dantzig-Wolfe decomposition combined with column generation is applied to compute the LP relaxation the time-indexed formulation of a single-machine scheduling problem. The LP relaxation is given by the optimal linear combination of generated pseudoschedules (columns). In [31], a pseudoschedule is a machine schedule in which a job can appear more than once. The approach of [31] has the advantage of polynomial time for computing pseudoschedules by means of Dijkstra’s algorithm since the column generation subproblem is modeled as a shortest path search. As we are interested in searching in reasonable time problem integer solutions, and not only the LP relaxation, in this work, we extend the germinal idea of [31] to the search of true schedules as a feasible selection of true schedules is an integer problem feasible solution. For this purpose, we introduce colored arcs in the problem graph at the price of upgraded problem complexity. The MRRPRC problem is an NP-Hard problem.

The authors of [17] resort to a column generation algorithm with an NP-Hard pricing problem for generating true schedules. There, such an NP-Hard pricing problem is solved through CP to generate as many optimal columns as possible at the price of relatively high computation times. We think that generating true schedules is winning for fast detecting high-quality integer problem solutions and that dynamic programming, in reason of potential computational efficiency, is more promising than CP that offers the mere advantage of generating a large number of optimal columns. Beyond these considerations, some of the dominance and optimality rules that are developed for improving the label-correcting algorithm we present in this section are derived from those developed by [17] for reducing the number of optimal columns generated at each call to CP.

Problem (35)-(45) can be decomposed by building a graph for each SG, which allows removing coloring Constraints (41) and (42); consequently,

the algorithm for the paths search is simpler. The search of the best path through the different SG graphs can be optimized by means of simple pruning strategies avoiding the full exploration of each SG graph; details are in Section 6.2.5. The description in this section is intended for solving the MRRPRC problem over a single SG graph.

Let a simplified notation be used for describing the algorithm: let set $I_{mj} \subseteq I$ be denoted (with an abuse of notation) with its membership set I . In this section, there is no need to differentiate. Surgeries provided as input of the RP algorithm are those that can be scheduled for day j and belong to SG m .

The algorithm that we developed to solve the MRRPRC is a label-correcting algorithm. Such an approach is an extension of the Ford-Bellman algorithm to take into account resource constraints and is quite common for vehicle routing problems with resource constraints, *e.g.*, papers [12] and [14] in the context of column generation. Labels depict paths and indicate the resource consumption of paths. Labels are generated throughout the algorithm and are sorted out as node attributes. Each node has its own set of labels that stands for the incoming paths that terminate with the node. Nodes in the graph are iteratively evaluated extending every incoming label toward every possible successor node. The algorithm terminates when no new labels are created. In Section 6.2.2, some dominance rules to limit the proliferation of labels are introduced. A detailed description of the algorithm is reported in Section 6.2.5.

The algorithm that we developed implements a *breadth-first search*, as do most labeling algorithms. Such a strategy has the benefit of a quick generation of a diversified set of paths searching the optimal one, but it can be longer than *depth-first search* in finding a path reaching the destination node. To overcome this potential weakness of the breadth-first search, a heuristic algorithm for extending any new label (path) until the destination node v_T is developed; the heuristic algorithm is described in Section 6.2.3. The computation of an upper bound for the best RP is explained in Section 6.2.4. With upper and lower bounds, pruning strategies can be implemented to further limit the proliferation of labels.

6.2.1. Label definition and extension function

Let L be the label standing for a path from the origin node v_0 to node v_t . A label is defined by the following attributes: t_L the terminal node time index, W_L the path revenue, C_L the path coloration, S_{aL} the consumption of each resource $a \in A$ and f_L the label infection level. Note that the infection level is also a resource for labels. The label consumption of resource f is increased or decreased according to the label extension: any time a path is extended with a surgery arc or an OR cleaning arc, the infection level f_L of the label is updated accordingly, *i.e.*, lifted, lowered or kept unchanged.

For the label extension function, with an abuse of notation, let index i denote not only a color $i \in I$ but also an OR cleaning $i \in F$ or an OR idling as well. A new label is created accordingly with the label extension function $Extend : L \rightarrow L'$. The label function, in the right-hand side, defines the attribute update operations of the label extension and the related constraints/conditions. In order, the constraints/conditions are: the revenue has to be maximized, the color of the extending arc has to not already belong to the label or the extending arc has to be OR cleaning or OR idling, there is no constraint for the time-index update, the remaining capacity of resource $a \in A$ has to be sufficient and the arc infection level has to be the greater or equal to that of the label if the extending arc is a surgery arc or, if it is OR cleaning, the infection level of the label to downgrade f_L has to equal that foreseen for the $OCT(f', f_i)$, *i.e.*, $f' = f_L$. Note that the label infection level remains unchanged if the extending arc is an OR-idling arc.

6.2.2. Dominance rules

The optimal solution of the MRRPRC problem can be found by considering for the label extension only nondominated labels (paths) and colors. In the following, we define some dominance rules to apply for limiting the proliferation of labels.

Proposition 2. *Given labels L and L' reaching the same node v_t , label L' dominates label L if $W_{L'} \geq W_L$, $C_{L'} \subseteq C_L$ and $f_{L'} \leq f_L$.*

PROOF. Let L'' be a complete extension of L to node v_T . The extension of L' to node v_T through the same surgery arcs extending L is feasible since all colors in $C_{L''} \setminus C_L$ are reachable for L' , *i.e.*,

$(C_{L''} \setminus C_L) \cap C_{L'} = \emptyset$, $f_{L'} \leq f_L$ and $S_{aL'} \leq S_{aL}$ for any $a \in A$. The revenue of the extended path from label L' equals $W_{L'} + W_{L''} - W_L$, and therefore, L' dominates L .

Let $w_{it}^{max} = \max\{w_{it} : t = t, \dots, T\}$ and $w_{it}^{min} = \min\{w_{it} : t = t, \dots, T\}$ be, respectively, the maximum and the minimum revenue among arcs of color i from node v_t to node v_T .

Proposition 3. *Given label L of node v_t and the two feasible extensions L' and L'' obtained extending L with an arc of color i' and i'' , respectively, label L' dominates label L'' if*

1. $w_{i't}^{min} > w_{i''t}^{max}$,
2. $p_{i'} + \widehat{OCT} \leq p_{i''}$ and
3. $d_{ai'} + d_{ai''} > Q_a - S_{atL} \forall a \in A : d_{ai'} > 0 \wedge d_{ai''} > 0$;

and color i'' is thus not optimal for extending label L . Value \widehat{OCT} is an upper bound on the additional pre/post OR cleaning time required if an arc of color i'' is substituted with an arc of color i' given the infection level f_L . Possible values for $\widehat{OCT} > 0$ are reported in Table 2².

PROOF. Let L^* be a complete extension of L'' to node v_T . According to condition 3, both colors i' and i'' cannot appear in the extension of L , and the extension of L' to node v_T through the same surgery arcs extending L'' is thus feasible since all colors in $C_{L^*} \setminus C_{L''}$ are reachable for L' , *i.e.*, $(C_{L^*} \setminus C_{L''}) \cap C_{L'} = \emptyset$, and $S_{aL'} \leq S_{aL''}$ for any $a \in A$. By condition 2, the surgery arc of color i'' can be replaced in label L^* with a surgery arc of color i' , by condition 1, the label revenue is improved to at least $W_{L^*} + w_{i't}^{min} - w_{i''t}^{max}$, and therefore, L' dominates L'' .

Proposition 4. *Given label L of node v_t , resource $a \in A$ and color set $\bar{I}_{aL} \subseteq I_a$ such that $\bar{I}_{aL} \cap C_L = \emptyset$ and $d_{ai} < Q_a - S_{aL}$ for every $i \in \bar{I}_{aL}$, given color $i \in \bar{I}_{aL}$ and set D_i of colors that dominate color i with respect to conditions 1 and 2, color i is not optimal for extending label L if it holds that*

4. $d_{ai} + \sum_{i' \in D_i \cap \bar{I}_{aL}} d_{ai'} > Q_a - S_{aL}$.

²In the table, $OCT(L, i')$ and $OCT(L, i'')$ are the OR cleaning times required to add a surgery arc, respectively, of color i' or i'' , given the label infection level f_L .

$$L' = \text{Extend}(L) = \begin{cases} W_{L'} = W_L + w_{it}, & \text{To maximize} \\ C_{L'} = C_L \cup i, & (i \notin C_L \wedge f_i \geq f_L) \vee i \notin I \\ t_{L'} = t_L + p_i, & - \\ S_{aL'} = S_{aL} + d_{ai}, & Q_a - S_{at} \geq d_{ai} \\ f_{L'} = f_i, & (i \in I \wedge f_i \geq f_L) \vee (i \in F : f' = f_L) \end{cases} \quad (49)$$

PROOF. Let L^* be a complete extension of L to node v_T that includes color i . Because of condition 4, L^* can never include the entire color set D_i , i.e., $D_i \setminus (C_{L^*} \cap D_i) \neq \emptyset$ always. Thus, by condition 2, the surgery arc of color can be replaced with an arc of a color $i' \in D_i \setminus (C_{L^*} \cap D_i)$ and, by condition 1, the revenue of label L^* is improved to at least $W_{L^*} + w_{i't}^{min} - w_{it}^{min}$. Color i is therefore not optimal for extending label L of node v_t .

Proposition 5. *Given label L and color set $\bar{I}_L \subseteq I$ such that $\bar{I}_L \cap C_L = \emptyset$ and $d_{ai} < Q_a - S_{aL} \forall i \in \bar{I}_L, a \in A$, given color $i \in \bar{I}_L$ and set D_i of colors that dominate color i with respect to conditions 1 and 2, color i is not optimal for extending label L if it holds that*

$$5. p_i + \sum_{i' \in D_i \cap \bar{I}_L} p_{i'} > T - t.$$

PROOF. Let L^* be a complete extension of L to node v_T that includes color i . Because of condition 5, L^* can never include the entire color set D_i , i.e., $D_i \setminus (C_{L^*} \cap D_i) \neq \emptyset$ always. Thus, by condition 2, the surgery arc of color can be replaced with an arc of a color $i' \in D_i \setminus (C_{L^*} \cap D_i)$ and, by condition 1, the revenue of label L^* is improved to at least $W_{L^*} + w_{i't}^{min} - w_{it}^{min}$. Color i is therefore not optimal for extending label L of node v_t .

6.2.3. Heuristic RP and best RP lower bound computation

Given label L that stands for a path reaching node v_t , the lower bound $LB(L)$ of L is computed by means of a greedy procedure. The path is iteratively extended with feasible surgery arcs as long as some arcs are selectable or the destination node v_T is not reached. The label extension function is defined by (49). At each iteration, the most promising arc for the label extension is selected according to

$$\arg \max_{(i, v_t)} \frac{w_{it}}{p_i} \cdot \left(\frac{w_{it}}{w_{it}^{max}} \right)^2, \quad (50)$$

i.e., the marginal revenue of selecting a surgery arc of color i for time t is pondered with the squared ratio of selecting a surgery arc of color i for time t to the optimal time selection of a surgery arc of color i . This rule for a greedy arc selection provides the best results among other possible rules we tested through numerical experiments.

If no surgery arc can be selected, the current label is extended with idle time arcs to reach v_T . Note that OR cleaning arcs are not considered in this heuristic.

6.2.4. Best RP upper bound computation

For the upper bound calculation of a label, we developed a method based on the well-known algorithm for the LP relaxation of the multidimensional 0-1 knapsack problem (MKP), see [13].

Given a label L and its infection level f_L , the LP solution of the MKP is computed for every infection level from $f_{min} = \min\{f_i : i \notin C_L\}$ to f_L . For every infection level $f \in [f_{min}; f_L]$, the knapsack to fill has a dimension for each resource (surgeon) $a \in A$ with capacity $Q_a - S_{aL}$ and a dimension for the residual distance to reach the destination node v_T with capacity $d(v_t, v_T) = T - t - OCT(L, f)$. To fill the multidimensional knapsack, there is an item for each color $i \notin C_L : f_i \geq f$ with volume equal to the arc length p_i and revenue equal to $w_{it'}^{max}$ with $t' = t + OCT(L, f)$ (the maximum arc revenue of arcs of color i from $v_{t'}$ to v_T). Let thus $UB_f(L)$ be the LP solution for MKP computed for label L and infection level f . The upper bound for label L is finally $UB(L) = \max\{UB_f(L) : f \in [f_{min}; f_L]\}$.

6.2.5. Description of the algorithm

Let $G_m : m \in SG$ be the graph of SG m . Let also $LB_0(G_m)$ and $UB_0(G_m)$ be, respectively, the lower bound and the upper bound calculated over graph G_m given the empty label L_0 . Upper bound $LB_0(G_m)$ and lower bound $UB_0(G_m)$ are valid bounds for the best rainbow path that can be found over graph G_m . The upper bound $UB_0(I) =$

$f_{i'} \text{ vs. } f_L$	$f_{i'} \text{ vs. } f_{i''}$	$f_{i''} \text{ vs. } f_L$	\widehat{OCT}
$f_{i'} < f_L$	$f_{i'} < f_{i''}$	$f_{i''} < f_L$	$OCT(L, i') - OCT(L, i'')$
$f_{i'} < f_L$	$f_{i'} < f_{i''}$	$f_{i''} > f_L$	$OCT(L, i') - OCT(i'', L)$
$f_{i'} > f_L$	$f_{i'} > f_{i''}$	$f_{i''} < f_L$	$OCT(i', L) - OCT(L, i'')$
$f_{i'} > f_L$	$f_{i'} > f_{i''}$	$f_{i''} > f_L$	$OCT(i', L) - OCT(i'', L)$
$f_{i'} < f_L$	$f_{i'} < f_{i''}$	$f_{i''} = f_L$	$OCT(L, i')$
$f_{i'} > f_L$	$f_{i'} > f_{i''}$	$f_{i''} = f_L$	$OCT(i', L)$

Table 2: Value for OCT bound \widehat{OCT}

$\max\{UB_0(G_m) : m \in SG\}$ is thus a valid upper bound on the best RP that can be found through the SG graphs created providing to the algorithm as input the set of surgeries I . In the following, we describe the algorithm for searching the best RP over SG graphs such that $UB_0(G_{m'}) \geq \max\{LB_0(G_m) : m \in SG\}$; other graphs are pruned.

Let Λ_t be the set of labels of node v_t (*i.e.*, the set of paths reaching and terminating with node v_t) and $\mathbf{\Lambda}$ be the set of all created labels. Let UB and LB be the global search upper and lower bound, respectively. Function $FAOE$ is a set filtering function retaining as output the set of arcs that are feasible and optimal for extending the label provided as input. The arc feasibility is defined by constraints of function (49) and the optimality by Propositions 3, 4 and 5 defined in Section 6.2.2. Function $Extend$ is the label extension function as defined with expression (49), and $LBproc$ is the heuristic RP algorithm described in Section 6.2.3. Function EFF is a set filtering function retaining as output the set of nondominated labels passed as input.

The RP algorithm evaluates graph nodes according to their topological order, *i.e.*, from the source node v_0 to the sink node v_T . Each node v_t is evaluated extending every nondominated label that reaches the node with any feasible outgoing arc. Labels with an upper bound lower than the best found RP that reaches the sink node v_T are pruned. The algorithm returns the set of nondominated RPs that start from node v_0 , reach node v_T , and cannot be further extended. Algorithm 1 provides the pseudocode of the RP algorithm.

In the context of column generation, the RP algorithm can be terminated early, as soon as the size $|\Lambda_T|$ of rainbow paths reaches a given value. We found to explore all the SG graphs with a strictly positive upper bound $UB_0(G_m) > 0$ (those that may generate some columns with a positive reduced cost) more efficient; this is even the case if the up-

Algorithm 1 Rainbow Path algorithm

```

1: procedure RP()
2:    $LB \leftarrow -\infty$ ;
3:    $UB \leftarrow +\infty$ ;
4:    $\Lambda_t = \{\emptyset\} \forall v_t \in V$ ;
5:    $\Lambda_0 = \{L_0\}$ ;
6:
7:   for  $v_t \in V : t = 0, \dots, T$  do
8:      $\Lambda_t \leftarrow$  set of  $v_t$  labels;
9:     for all  $L \in \Lambda_t$  do
10:      if  $UB(L) > LB$  then
11:         $E_L^* \leftarrow \mathbf{FAOE}(L, E_{v_t})$ ;
12:        for all  $(i, v_t) \in E_L^*$  do
13:           $L' \leftarrow \mathbf{Extend}((i, v_t), L)$ ;
14:           $L'' \leftarrow \mathbf{LBproc}(L')$ ;
15:           $LB \leftarrow \max\{LB; W_{L''}\}$ ;
16:           $\Lambda_{t'} \leftarrow \mathbf{EFF}(\Lambda_{t'} \cup L')$ ;
17:           $\Lambda_T \leftarrow \mathbf{EFF}(\Lambda_T \cup L'')$ ;
18:         $\Lambda_t \leftarrow \Lambda_t \setminus \{L\}$ ;
19:       $UB \leftarrow \max\{UB(L) : L \in \mathbf{\Lambda}\}$ ;
20:
21: return  $\{L \in \Lambda_T : W_L \geq LB\}$ ;

```

per bound $UB_0(G_m)$ is dominated by an already found lower bound $LB_0(G_{m'}) : m' \neq m$; the column generation is otherwise longer. In Algorithm 1, we omit implementation details about the column generation condition for termination and the search through the SG graphs to facilitate pseudocode clarity and readability as much as possible.

7. Computational Results

In this section, we validate through a set of computational experiments the effectiveness of the algorithms that we developed, namely, the basic column generation, the column generation with Benders' cuts and the branch-and-price-and-cut. We test the algorithms with a set of instances that we generated based on data from the CHUSE (University Hospi-

tal of Saint-Etienne, France). The instance generation procedure is described in Section 7.1. We also test the algorithms with the instances used in [17]. The CHUSE database tracks the surgical activity of several years.

Through the experiments, we first show the efficiency of the label-correcting algorithm used for solving the column generation pricing problems; the algorithm is described in Section 6.2. The effectiveness of the Benders’ cutting procedure developed for tightening the LP relaxation of the problem computed by column generation is then evaluated, and the quality of the branch-and-price-and-cut algorithm for the search of integer problem solutions is compared with results of [17]. We also benchmarked our branch-and-price-and-cut with the results obtained with a commercial IP solver (CLPEX).

All the algorithms are coded in C++ and resort to the IBM ILOG CPLEX APIs (version 12.8) for LP and ILP routines. All the experiments are run with a processor *Intel Xeon E7-8890 v3* at 2.50 GHz.

7.1. Instances

In all the instances that we generated, the planning horizon is five days (a regular work week) and duration and times are expressed in time slots of 5 minutes. In the CHUSE database, the surgery duration, the surgical specialty and the number of nurses that assisted a surgery are known for every surgery. The CHUSE database records do not provide all the problem data; therefore, we complemented database records with randomly generated data. Even though not tracked in the CHUSE database, missing problem data are available information at the problem decision epoch.

For every surgery, the release date is drawn uniformly in the interval $[0; |J| - 1]$ and the due date is drawn uniformly in the interval $[ReleaseDate; 14]$. This rule follows the suggestion of [17] and [10], the authors of the two papers generate their instances drawing surgery due dates uniformly in the interval $[1; 14]$. Two infection types are considered, infectious and noninfectious; the infectious type is randomly assigned to surgeries with probability $\frac{1}{2}$. A surgery can be normal or expensive. The average reimbursement is estimated, in France, at €3,375.00 for normal surgeries and at €7,830.00 for expensive surgeries. The expensive type is randomly assigned to surgeries with probability equal to a ratio

we arbitrarily defined for every surgical specialty. The surgery reimbursement is drawn from a normal distribution with a mean equal to the reimbursement average value for the surgery type and standard deviation equal to $\frac{2}{3}$ of the mean (because the standard deviation is unknown); values lower than €500.00 and greater than €24,000.00 are discarded; we choose the normal distribution and the limit values for the surgery reimbursement arbitrarily. Values of the ratio of expensive surgeries are reported in Table 3 for each surgical specialty, column *e.s. ratio*. In Table 3, we also reported the surgical group of the specialty (*SG*), the average surgery duration for the specialty (*ave.*), the percentage of surgeries with a duration lower than 60 minutes (< 60), lower than 90 minutes (< 90) and lower than 150 minutes (< 150), and the number of surgeries in the database for each surgical specialty.

Inputs for the instance generation are as follows: the desired number of surgeries, the desired number of surgeons, the selected SGs (then the surgical specialties), and the CHUSE database. It holds that the number of surgeons is always greater than the number of surgical specialties related to the selected SGs.

To generate an instance, a surgeon can be assigned to one and only one surgical specialty, one surgeon is assigned to every surgical specialty until all specialties have one surgeon assigned, and the remaining surgeons are then assigned to a specialty randomly selected. The number of days that a surgeon is available in the planning horizon is drawn from a triangular distribution with $min. = 1$, $max. = 5$ and $mode = 1$, the weekdays (from Monday to Friday) when the surgeon is available are randomly selected, and the surgeon maximum daily time is drawn for every selected day from a triangular distribution with $min. = 4$, $max. = 8$ and $mode = 4$. Parameters of the two triangular distributions are estimated using the data available in the CHUSE database.

Surgeries are selected by iterating over surgeons. At each iteration, a given number of surgeries belonging to the surgeon specialty are randomly selected in the CHUSE database; this selection of surgeries is repeated until the desired number of surgeries belong to the instance. The number of surgeries to select at each iteration is calculated, for every surgical specialty, as 1 plus the rounding of the ratio between the average surgery duration multiplied by a scalar greater than 1 and the average surgery duration of the given specialty. This

SG	specialty	e.s. ratio	Duration			surgeries	
			ave.	< 60	< 90		< 150
OM	maxillofacial	65.00	101.4	5.47	46.42	88.83	5937
OM	otolaryngology	15.00	103.9	4.86	41.29	88.31	10372
NR	neurosurgery	85.00	134.8	0.72	15.76	65.13	9525
ORTR	bone	50.00	112.5	2.61	34.02	81.73	18901
CV	cardiovascular	75.00	133.8	0.87	20.85	64.74	7156
INF	pediatric	50.00	103.5	4.81	44.12	87.22	11326
DIG	alimentary canal	20.00	121.7	2.35	29.25	73.78	8185
GYURA	urology	20.00	97.6	6.16	50.58	91.11	6847
GYURA	gynecology	20.00	113.0	4.49	35.48	80.20	6973
GTPOT	oncological pneumology	40.00	83.9	8.99	62.23	99.32	1480
GTPOT	general and thoracic surgery	45.00	117.4	2.80	30.76	77.93	12692
AMB	daily	5.00	86.6	8.89	59.56	98.41	7275
OPT	ophtalmonlogy	15.00	92.7	6.45	50.86	96.42	7292

Table 3: Expensive surgery ratio for specialties

value is larger for surgical specialties with a prevalence of short surgeries and smaller for specialties with a prevalence of long surgeries; the scope of this value is to allow the generation of more balanced instances in terms of total surgery time of the different surgical specialties.

The number of nurses belonging to each SG m is computed as

$$\max \left\{ \frac{\sum_{i \in I_m} p_i n_i}{|J| - 1} \cdot T_m, \max\{n_i : i \in I_m\} \right\} \quad (51)$$

where T_m is the regular daily work time for nurses belonging to SG m . One day off, randomly selected in the planning horizon, is assigned to every nurse. For the other days, the cut off time of the regular work time is calculated as the OT opening time (conventionally 0) plus T_m . The value of T_m of 480 minutes is set for every $m \in SG$.

We generated three instance sets, denoted with capital letters S, M and H. For instances of set S, the selected surgeries belong to two SGs, *OM* and *OPT*, surgery durations are in the interval [40 minutes, 150 minutes] and the number of surgeries belonging to an instance is 60, 80 or 100; three surgeons and two equivalent ORs are available for surgery. For instances of set M, the selected surgeries belong to four SGs, *OM*, *OPT*, *NR* and *INF*, surgery durations are in the interval [40 minutes, 240 minutes] and the number of surgeries is from 60, 80 or 100; nine surgeons and six equivalent ORs are available for surgery. For instances of set H, surgeries belong to ten SGs (all considered SGs) and there are

160 surgeries with durations in the interval [40 minutes, 240 minutes]; twenty-four surgeons and twelve equivalent ORs are available for surgery. We generated five instances for each problem setting in S and M, and ten instances for set H.

The selection of SGs of sets S and M is driven by the purpose of designing a stress test for the developed algorithms, given that, as reported in Section 7.3, instances with only short surgeries are more difficult to solve. We verified with preliminary numerical experiments that other possible (equivalent) selections of SGs give similar results and lead to the same conclusions. For instances in S and M, we generated instances with an OR capacity lower than the expected total surgery time calculated as the product of the average surgery duration with the number of surgeries. The number of surgeons is chosen to obtain a tight total surgeon capacity with respect to the OR capacity. The 65 instances used in [17] are divided into three sets — A, B and C — reflecting different situations. For all these instance sets, there are no nurses, the revenue of each surgery is equal to the surgery duration and the OT management policy is open block, no SGs have to be assigned to ORs (there is virtually only one SG). In set A, surgeons are the more binding resources for scheduling the surgeries, and instances count from 40 up to 120 surgeries with only medium-long durations from 120 minutes up to 240 minutes. Instance set B differs from set A only with respect to the number of ORs open each day; in this set of instances, the OR capacity is also binding. In set

C, both surgeons and ORs are binding resources, the number of surgeries is from 60 up to 100 and surgery durations are in the interval [40 minutes, 150 minutes]. For further details, see [17].

7.2. Parameters

In the experiments we ran, we put a time limit of 3 hours for the total time of each run and a time limit of 2 minutes for the LB computing (solving the restricted master problem with integer variables). The target number of columns to add to the RMP at each iteration is defined by a function of the number of surgeries in the instance. This function is $\max\{4; 2^{7-0.05|I|}\}$ and, for the values 40, 60, 80, 100, 120 and 160 of $|I|$, produces the series $\{32, 16, 8, 4, 4, 4\}$. All the experiments are run as single-thread processes; there is no parallelism for both our code and calls to the commercial solver.

7.3. Results

In this section, we report numerical results for the experiments we ran for the 40 instances that we generated and the 65 instances by [17]. In all the tables we present, the results for instances of sets A, B and C are reported as average values. This is not only because the number of instances is very large but also because it provides a quick comparison with the results reported in [17], that are reported as average values.

7.3.1. Instance benchmarking: constructive heuristic algorithm

To test the effective hardness of evaluated instances, we developed a simple constructive heuristic algorithm. The heuristic algorithm keeps a priority queue of unscheduled surgeries, the surgery priority is assigned according to the early due date rule and, for equal due dates, surgeries with higher revenue have higher priority. Days in the planning horizon are evaluated in a rolling horizon fashion, and surgeries are scheduled greedily by processing the priority queue. Given the surgery ahead of the priority queue, the first starting time that fits the required capacity of the assigned surgeon, OR and required nurses is selected for scheduling the surgery, the concerned resource capacity is blocked from the surgery starting time to the surgery completion time and the surgery is removed from the

queue. A surgery cannot be scheduled before its release date or after its due date.

The solution quality of the simple heuristic algorithm that we developed to benchmark the hardness of the instances is by far inferior to that of the exact optimization algorithms we devised. The heuristic dramatically fails to find a feasible solution that schedules the whole set of mandatory surgeries for many instances; this happens for 80% of instances of set S and for 100% of instances of all the other sets, *i.e.*, sets M, H, A, B and C. For instances of set S, the average gap of the heuristic solution to the best known integer solution is 43%, and the minimum and the maximum are 25% and 58%, respectively. For instances of set M, the average gap is 22%, the minimum 9% and the maximum 32%. For instances of set H, the average gap is 31%, the minimum 20% and the maximum 44%. For instances of set A, B and C, respectively, the average gap is 12%, 10% and 3%, the maximum 20%, 19% and 13% and the minimum gap is 6% for set A and 0% for both sets B and C.

7.3.2. Basic CG

In Table 4, the basic CG algorithm is evaluated. We reported the LP relaxation computed by CG (*rev. LP*), the solution of the ORRMP with integer column variables (*rev. IP*) and the percentage gap of the IP solution with respect to the LP relaxation (*gap*). We also reported the time of the basic CG algorithm (*time LP*), the time spent solving the ORRMP (*time MP*), the time spent for the CG, *i.e.*, for solving the pricing problems, (*time CG*) and the time spent solving the ORRMP with integer column variables (*time IP*). We finally report the iteration count of the basic CG algorithm (*iter. CG*) and the number of columns added to the ORRMP (*columns*). For instance sets A, B and C, in column *Rev. IP*, we reported between parenthesis the number of instances for which an integer solution is found by the algorithm; each row shows the average value for the given set.

Table 4 provides evidence that, for instances with only short surgeries (set S), most of the computation time is spent for the CG. This CG time decreases for instances with also longer surgeries, sets M and H. For set M, the time spent for the ORRMP solution and the CG is relatively balanced (the same order of magnitude). For set H, the time spent for the CG is significantly smaller than the time spent solving the ORRMP, in most of the cases

Instance	rev. LP	rev. IP	gap (%)	time LP [sec.]	time MP [sec.]	time CG [sec.]	time IP [sec.]	iter. CG	columns
S60-1	117618	113604	3.41	19.2	4.5	14.4	0.1	95	752
S60-2	159934	149562	6.49	94.2	58.7	35.6	0.9	896	3141
S60-3	149045	145260	2.54	299.7	36.8	262.9	0.3	309	1817
S60-4	161093	161093	0.00	24.9	2.9	21.7	0.0	108	591
S60-5	111489	104192	6.55	2.1	0.5	1.3	0.0	31	214
S80-1	171743	170226	0.88	238.7	122.3	116.4	1.0	1355	3072
S80-2	145758	143487	1.56	140.5	19.4	121.1	0.1	306	1227
S80-3	136392	129849	4.80	35.9	13.6	22.4	0.2	318	956
S80-4	138825	121656	12.37	21.1	3.7	17.3	0.1	108	547
S80-5	160769	157079	2.30	1599.2	166.4	1432.7	1.3	803	2748
S100-1	148556	132610	10.73	270.3	21.8	248.5	0.1	727	1192
S100-2	154182	152521	1.08	502.0	24.8	477.2	0.1	493	960
S100-3	75948	73556	3.15	10.8	1.3	9.5	0.0	176	216
S100-4	168372	164986	2.01	897.0	130.6	766.4	0.2	1598	2183
S100-5	169078	160515	5.06	477.6	10.7	466.8	0.1	402	721
M60-1	179959	179489	0.26	5.4	2.1	3.3	0.1	91	413
M60-2	224376	213885	4.68	58.9	39.2	19.7	2.8	401	1561
M60-3	181254	180460	0.44	4.3	1.3	2.9	0.0	46	262
M60-4	220481	211937	3.88	37.5	21.7	15.7	0.5	316	1315
M60-5	227003	209209	7.84	9.4	3.6	5.8	0.0	131	462
M80-1	257464	235575	8.50	19.6	10.7	8.9	0.1	161	803
M80-2	206327	187989	8.89	112.8	42.2	70.6	1.1	771	1551
M80-3	236646	227641	3.81	144.7	67.4	77.3	1.0	1016	2001
M80-4	284881	280529	1.53	64.0	30.7	33.3	0.4	646	1334
M80-5	179000	145472	18.73	471.7	258.2	213.5	9.0	1126	3647
M100-1	296298	280485	5.34	481.4	266.3	215.0	1.5	1591	2957
M100-2	292928	277942	5.12	90.8	43.8	47.1	0.4	1056	1313
M100-3	263210	242692	7.80	151.9	84.3	67.6	0.6	946	1583
M100-4	216244	193964	10.30	212.0	130.3	81.7	0.7	1181	1848
M100-5	281826	268728	4.65	131.5	73.4	58.1	0.2	906	1375
H160-1	630674	593505	5.89	1186.2	1112.9	73.4	23.5	406	11286
H160-2	605497	558853	7.70	326.6	254.7	71.9	9.6	161	7749
H160-3	588978	542703	7.86	1391.6	991.7	399.9	94.3	2001	4867
H160-4	711602	667348	6.22	404.7	342.9	61.8	4.3	246	7597
H160-5	627127	577052	7.98	298.6	222.1	76.5	3.3	196	4343
H160-6	529833	496900	6.22	712.2	414.8	297.4	109.2	86	12255
H160-7	499528	459498	8.01	68.6	51.7	16.9	0.4	76	1514
H160-8	601766	574734	4.49	54.2	39.0	15.2	0.5	76	1133
H160-9	549814	537190	2.30	246.3	195.9	50.3	2.8	136	4554
H160-10	525786	506039	3.76	148.8	119.2	29.6	2.0	181	4405
Set									
A40	1431	1394(5)	2.59	3.4	1.8	1.2	0.3	47	641
A60	2020	1859(5)	7.96	24.9	18.3	6.6	37.3	143	1170
A80	2307	2104(5)	8.80	334.7	203.2	131.5	22.2	761	2605
A100	2401	2103(5)	12.41	658.5	372.8	285.7	68.8	1231	2820
A120	2423	2161(4)	11.03	521.3	261.5	259.8	18.2	948	2314
B40	1371	1355(5)	1.12	6.2	3.7	2.4	5.4	37	762
B60	1866	1824(5)	2.28	61.4	22.0	39.4	14.2	131	1274
B80	2205	2096(5)	4.92	183.6	105.6	78.0	56.6	344	1941
B100	2416	2176(5)	9.87	1532.8	673.3	859.6	108.6	1956	3788
B120	2480	2172(5)	12.40	1630.5	568.4	1062.1	108.1	1657	3526
C60	576	500(5)	13.19	39.1	16.8	22.4	2.8	74	1136
C80	792			35.0	17.8	17.2	0.2	102	780
C100	978			13.4	8.5	5.0	0.1	123	576

Table 4: Evaluation of the basic CG Algorithm

smaller by one order of magnitude; this is because of the longer CG convergence due to the large number of surgeries and ORs in the instances of set H.

The same relation between the time of the ORRMP solution and the CG time is evidenced through the results for sets A, B and C.

The average computation time for a call to CG can be calculated, for each instance, by dividing the *time CG* by *iter. CG*. As expected, the shorter the surgeries are, the longer this average time for a CG call is. The comparison between set S and M is emblematic. The average of this average time for a call to CG is 0.45 seconds for instances of set S and 0.07 seconds for instances of set M. The maximum average time for a call to CG is, respectively, 1.78 seconds for instances of set S, 0.19 seconds for instances of set M.

The gap of the IP solution with the LP relaxation of the basic CG algorithm seems not to be influenced by the instance characteristics of size and surgery durations for the instances that we generated: the gap is in the range of 0.0% and 18.7%. For sets of instances by [17] — A, B and C — the gap is in the range of 0.3% and 15.5%.

The number of columns generated is quite low for every set of instances. It is reasonably larger for set H, with many surgeries and many ORs available. It is also larger for instances of sets A, B and C, as expected, because of the open block strategy.

7.3.3. CG with Benders cuts

In Table 5, the CG with the Benders' cuts algorithm is evaluated. As in Table 4, we report values of the LP relaxation, of the integer solution and of the relative gap: (*rev. LP*), (*rev. IP*) and (*gap*), respectively. We report the total time of the CG-with-Benders'-cuts algorithm spent for computing the LP relaxation (*time LP*); this time includes the time spent solving the SSP (*time SSP*) and the NSP (*time NSP*). We finally report the total number of columns added to ORRMP (*columns*), the number of cuts generated by solving the SSP and the NSP, (*cuts SSP*) and (*cuts NSP*), respectively. As in Table 4, for instance sets A, B and C, in column *Rev. IP*, we report between parenthesis the number of instances for which an integer solution has been found by the algorithm.

The results in Table 5 show the efficacy of Benders' cuts in closing the IP optimality gap by improving the quality of the LP relaxation. For most of the small instances, sets S and M, the optimality of the IP solution is proven by applying the Benders' cutting procedure. For large instances (set H), the quality of the LP relaxation is significantly improved by Benders' cuts, but there are no IP solutions with proven optimality.

Although a remarkable improvement of the LP

relaxation is attained, between 0% and 14% with an average of 3%, the Benders' cutting procedure requires a computation time that is not negligible. For instances of set S, the average time spent for solving the SSP and the NSP is, respectively, 16% and 13% of the total CG-with-Benders'-cuts time, 6% and 7% for instances of set M, respectively, and 2% (for both SSP and NSP) for instances of set H.

7.3.4. Branch-and-price-and-cut

Table 6 reports results for the IP commercial solver (CPLEX) and the branch-and-price-and-cut algorithm; the table is divided into two subtables, (*IP Solver*) and (*Branch-and-Price-and-Cut*). We reported, for both the IP solver and the B&P&Cut algorithm, the LP relaxation and the IP solution that result at the end of the IP optimization, (*rev. LP*) and (*rev. IP*), respectively. For the IP solver, we reported the percentage gap (*gap*) of the IP solution with respect to the LP relaxation and the optimization time (*time IP*). For the B&P&Cut algorithm, we reported the percentage gap (*gap*), the optimization time at which the gap is reached (*time gap*) and the total optimization time (*total time*) that also includes the total time spent solving the ORRMP with integer column variables (*time IP*). For instance of sets A, B and C, in the column *Rev. IP*, for both the *IP solver* and the *branch-and-price-and-cut*, we reported between parenthesis the number of instances for which an integer solution has been found by the algorithm.

Table 6 reveals that the branch-and-price-and-cut algorithm that we developed outperforms the IP commercial solver. Our algorithm is capable of finding the same or a better integer solution and stops the optimization because of the tight problem bounding provided by the computed LP relaxation. The IP commercial solver fails to identify an integer solution for large instances — set H, A120, B100 and B120 — and its performance decreases as the instance size (number of surgeries) increases, *i.e.*, very poor quality integer solutions and optimality gaps significantly larger than those found by the branch-and-price-and-cut. Computation times for the IP commercial solver are of the same order of magnitude of the branch-and-price-and-cut over instances of sets S and M. Computation times of the IP commercial solver grow significantly as the number of ORs available for the surgery scheduling increases.

Most of the improvement of the LP relaxation is

Instance	rev. LP	rev. IP	gap (%)	time LP [sec.]	time SSP [sec.]	time NSP [sec.]	time IP [sec.]	columns	cuts SSP	cuts NSP
S60-1	114168	114168	0.00	46.6	10.8	3.8	0.1	762	8	0
S60-2	149562	149562	0.00	238.0	2.5	6.0	1.7	4328	4	4
S60-3	147640	145260	1.61	377.2	26.7	15.2	0.4	1859	4	0
S60-4	161093	161093	0.00	27.6	1.3	1.3	0.0	591	0	0
S60-5	104192	104192	0.00	50.9	19.9	21.0	0.0	225	12	9
S80-1	170226	170226	0.00	473.4	23.2	18.2	2.1	3801	3	1
S80-2	143487	143487	0.00	372.4	11.6	8.0	0.4	1513	3	3
S80-3	129849	129849	0.00	240.2	35.9	166.4	0.2	957	2	0
S80-4	121656	121656	0.00	42.8	12.0	9.0	0.1	547	1	0
S80-5	158740	157079	1.05	1844.4	27.6	2.3	2.4	3306	10	3
S100-1	132610	132610	0.00	815.2	375.4	88.1	0.2	1228	7	1
S100-2	152521	152521	0.00	1381.8	246.4	218.6	0.3	1057	22	21
S100-3	73613	73556	0.08	44.4	12.5	4.5	0.1	262	7	0
S100-4	164986	164986	0.00	1712.2	68.1	54.4	0.4	2342	4	4
S100-5	160515	160515	0.00	597.9	116.6	11.7	0.1	723	1	1
M60-1	179489	179489	0.00	39.5	9.4	3.0	0.1	419	28	0
M60-2	215312	213885	0.66	191.5	2.6	3.8	13.3	1854	16	13
M60-3	180460	180460	0.00	8.9	1.2	1.6	0.0	271	2	0
M60-4	211937	211937	0.00	56.1	0.7	1.1	0.8	1813	2	1
M60-5	209209	209209	0.00	19.7	2.4	2.3	0.1	469	4	3
M80-1	241332	235830	2.28	52.3	9.8	4.3	0.1	854	8	2
M80-3	190041	187989	1.08	229.1	8.5	10.6	2.4	1944	6	5
M80-3	227641	227641	0.00	258.3	10.8	13.9	1.3	2117	14	5
M80-4	280529	280529	0.00	93.2	1.7	3.4	0.6	1632	4	5
M80-5	154405	145472	5.79	679.3	3.2	17.3	3.3	3957	3	5
M100-1	280485	280485	0.00	1214.9	8.1	39.1	2.9	3802	4	5
M100-2	277942	277942	0.00	162.5	8.2	29.5	0.6	1335	9	8
M100-3	243422	242692	0.30	208.8	3.7	10.0	0.9	1738	3	3
M100-4	193964	193964	0.00	200.3	3.9	5.4	0.6	1911	7	4
M100-5	268728	268728	0.00	192.1	3.3	6.1	0.4	1706	7	6
H160-1	605472	593505	1.98	3445.0	13.1	24.0	44.9	12653	15	13
H160-2	589814	560313	5.00	2617.7	17.5	32.3	36.3	11493	18	16
H160-3	549797	542703	1.29	2848.4	22.1	55.9	47.4	5031	29	28
H160-4	677788	669817	1.18	3151.8	16.8	33.5	55.4	11147	21	19
H160-5	605431	577052	4.69	1175.1	19.6	38.0	12.9	6725	12	14
H160-6	501733	494598	1.42	10635.6	27.5	28.1	110.1	12329	55	36
H160-7	483005	459498	4.87	637.9	32.8	28.9	1.0	2115	36	8
H160-8	586054	574734	1.93	121.4	10.8	7.2	0.6	1160	9	5
H160-9	539154	537190	0.36	10880.5	71.8	71.9	40.2	7894	105	29
H160-10	521027	506039	2.88	1678.3	14.5	14.0	36.6	7183	25	20
Set										
A40	1407	1394(5)	0.94	4.9	0.6	0.0	0.5	642	1	-
A60	1964	1876(5)	4.43	48.9	3.3	0.0	42.3	1321	4	-
A80	2198	2113(5)	3.82	660.9	18.4	0.0	53.6	3420	19	-
A100	2321	2180(5)	6.07	2632.5	33.5	0.0	109.1	5227	24	-
A120	2364	2201(4)	7.98	3377.2	75.5	0.0	67.5	5539	31	-
B40	1365	1355(5)	0.68	9.1	0.6	0.0	4.7	776	2	-
B60	1859	1824(5)	1.91	90.3	2.9	0.0	25.2	1323	4	-
B80	2143	2093(5)	2.31	574.1	21.7	0.0	76.5	2859	19	-
B100	2389	2222(5)	6.93	6101.5	58.3	0.0	117.5	6666	29	-
B120	2399	2223(5)	7.33	9937.2	112.5	0.0	117.9	8010	60	-
C60	576	500(5)	13.19	40.6	0.9	0.0	2.8	1136	0	-
C80	792			36.3	1.0	0.0	0.2	780	0	-
C100	978			16.2	1.0	0.0	0.1	576	0	-

Table 5: Evaluation of the Benders' cutting procedure

given by the Benders' cuts added at the root node (see Table 5), but the branch-and-price-and-cut is even effective for further closing the IP optimality

gap in both the directions, *i.e.*, tightening the LP relaxation and improving the IP solution quality. Such further gap closing for the LP relaxation is not

Instance	IP Solver				Branch-and-Price-and-Cut					
	rev. LP	rev. IP	gap (%)	time IP [sec.]	rev. LP	rev. IP	gap (%)	time gap [sec.]	time total [sec.]	time IP [sec.]
S60-1	114168	114168	0.00	23.3	114168	114168	0.00	46.4	46.4	0.1
S60-2	149562	149562	0.00	57.8	149562	149562	0.00	240.3	240.3	1.7
S60-3	145260	145260	0.00	441.4	145260	145260	0.00	431.7	431.7	0.9
S60-4	161093	161093	0.00	27.2	161093	161093	0.00	27.8	27.8	0.0
S60-5	104192	104192	0.00	37.9	104192	104192	0.00	62.7	62.7	0.0
S80-1	170226	170226	0.00	68.9	170226	170226	0.00	471.1	471.1	2.1
S80-2	143490	143487	0.00	309.7	143487	143487	0.00	296.3	296.3	0.4
S80-3	129849	129849	0.00	11.4	129849	129849	0.00	240.6	240.6	0.2
S80-4	121656	121656	0.00	209.7	121656	121656	0.00	41.9	41.9	0.1
S80-5	158182	157079	0.70	10800.1	157079	157079	0.00	3842.0	3842.0	11.2
S100-1	132610	132610	0.00	200.9	132610	132610	0.00	660.4	660.4	0.2
S100-2	152521	152521	0.00	456.1	152521	152521	0.00	1383.3	1383.3	0.3
S100-3	73559	73556	0.00	62.3	73556	73556	0.00	69.0	69.0	0.2
S100-4	164986	164986	0.00	203.8	164986	164986	0.00	1709.4	1709.4	0.4
S100-5	160515	160515	0.00	175.8	160515	160515	0.00	597.2	597.2	0.1
M60-1	179489	179489	0.00	74.4	179489	179489	0.00	31.7	31.7	0.1
M60-2	213885	213885	0.00	550.6	213885	213885	0.00	215.9	215.9	15.5
M60-3	180460	180460	0.00	94.0	180460	180460	0.00	9.0	9.0	0.0
M60-4	211937	211937	0.00	225.7	211937	211937	0.00	72.3	72.3	1.1
M60-5	209209	209209	0.00	74.8	209209	209209	0.00	19.9	19.9	0.0
M80-1	235830	235830	0.00	137.2	235830	235830	0.00	61.0	61.0	0.3
M80-2	188002	187989	0.01	9536.3	187989	187989	0.00	209.5	209.5	3.2
M80-3	227641	227641	0.00	728.6	227641	227641	0.00	198.6	198.6	1.0
M80-4	280529	280529	0.00	133.7	280529	280529	0.00	121.7	121.7	0.7
M80-5	154405	154405	0.00	3657.1	154405	154405	0.00	878.9	878.9	17.3
M100-1	280485	280485	0.00	557.8	280485	280485	0.00	1259.3	1259.3	2.9
M100-2	277942	277942	0.00	1761.1	277942	277942	0.00	162.5	162.5	0.6
M100-3	242692	242692	0.00	666.9	242692	242692	0.00	234.7	234.7	2.5
M100-4	193964	193964	0.00	986.6	193964	193964	0.00	276.6	276.6	0.7
M100-5	268728	268728	0.00	453.9	268728	268728	0.00	242.7	242.7	0.5
H160-1					605433	593505	1.97	4037.1	10855.5	1155.8
H160-2					588932	560313	4.86	10748.2	12168.7	372.7
H160-3					548329	542703	1.03	4320.8	10815.7	3032.2
H160-4					677788	669817	1.18	3161.3	10822.4	1952.3
H160-5					605431	577052	4.69	1117.4	10831.5	1735.4
H160-6					501733	494598	1.42	10787.2	11093.4	235.7
H160-7					483005	459498	4.87	386.9	10813.6	509.9
H160-8					586054	583984	0.35	9160.9	10811.8	554.8
H160-9					539158	537190	0.37	10861.5	10861.5	40.5
H160-10					520908	509640	2.16	3569.8	10847.0	2385.8
A40	1403	1403(5)	0.00	692.8	1403	1403(5)	0.00	102.7	102.7	18.9
A60	1965	1960(5)	0.25	6559.3	1962	1944(5)	0.89	7859.6	10807.2	7740.2
A80	2241	1539(5)	31.34	12709.7	2196	2135(5)	2.73	5731.4	10823.3	5737.3
A100	2287	1339(1)	41.45	35356.5	2321	2196(5)	5.34	7181.0	10902.5	4937.0
A120					2364	2207(5)	6.61	7874.9	10997.1	2981.1
B40	1364	1361(5)	0.23	4682.8	1361	1360(5)	0.07	477.9	4698.4	762.7
B60	1849	1823(5)	1.37	8273.8	1854	1835(5)	1.00	5245.3	10809.3	4959.4
B80	2191	1376(5)	37.23	16244.4	2142	2121(5)	0.96	3877.9	10837.8	7233.4
B100					2387	2247(5)	5.83	8530.6	10925.6	2403.9
B120					2399	2242(5)	6.54	11129.5	11205.2	453.7
C60	576	576(5)	0.00	3190.6	576	574(5)	0.31	6592.1	9570.6	3371.4
C80	792	760(5)	4.02	11987.1	792	686(5)	13.34	4400.6	12217.4	2787.8
C100	978	360(5)	63.16	19969.2	977	733(4)	24.31	7585.7	10803.8	1333.4

Table 6: Evaluation of the Branch-and-Price-and-Cut Algorithm

greater than the 2.28% over all the sets of instances, and for the integer solution, it can be even 13%.

In Table 7, we report some further details of the

branch-and-price-and-cut algorithm, but only average values for each set of instances. These are as follows: the number of columns added to the

ORRMP (*columns*), the iteration count (*iter.*), the number of created nodes (*nodes*) and the number of closed nodes (*nodes closed*).

Set	columns	cuts SSP	cuts NSP	iter.	nodes	nodes closed
S60	1553	6	3	1	1	1
S80	2114	5	2	2	2	1
S100	1122	9	5	1	1	1
M60	965	10	3	1	1	1
M80	2181	7	4	3	3	1
M100	2098	6	5	1	1	1
H160	11502	13	8	117	86	0
A40	739	1	-	68	131	1
A60	2189	11	-	637	1245	0
A80	3767	24	-	346	674	0
A100	5734	49	-	85	166	0
A120	5816	51	-	110	211	0
B40	9773	4	-	607	1045	2
B60	2588	42	-	799	1563	0
B80	3339	45	-	347	681	0
B100	7058	43	-	24	48	0
B120	8037	62	-	5	10	0
C60	4231	0	-	239	463	1
C80	4372	0	-	677	1353	0
C100	2957	0	-	1186	2367	0

Table 7: Evaluation of the Branch-and-Price Algorithm

7.3.5. Assessing the value of the hybrid open block strategy for surgeons

It may be argued that the OT policy scheduling that does not consider an OR-to-surgeon assignment on a daily basis (at least) is not so common and unreasonably upgrades the problem complexity; note that we define the open block strategy as "hybrid" because, in our model, surgeons are constrained to use only ORs assigned to their SG. To step forward and better explain why we considered this OT scheduling policy, we remark that: (1) the policy is adopted by CHU hospital, (2) the policy is considered in [17] and (3) the numerical experiments we ran show that there is a significant gain in allowing more than one surgeon to use the same OR on the same day and for surgeons to be able to use more than one OR on the same day.

By enforcing OR-to-surgeon assignment constraints, we revealed the following. For instances of set S, one instance of fifteen becomes infeasible, for six instances the OR-to-surgeon assignment has no impact (the optimal solution value does not change) and the remaining eight instances lose on average 2.91% of the revenue, where the minimum and the maximum loss of revenue are 0.49% and 9.04%, respectively. For instances of set M, one in-

stance of fifteen becomes infeasible, ten instances are not impacted and the remaining four instances lose on average 4.17%, where the minimum and the maximum loss of revenue are 1.52% and 6.21%, respectively. For instances of set H, three instances of ten become infeasible and for four instances the loss of revenue is on average the 1.25%, the minimum and the maximum losses are 0.34% and 2.39%, respectively. For instances of set A, there are nine instances of twenty-five that become infeasible and fifteen instances lose on average 3.38% of the revenue, where the minimum and the maximum losses of revenue are 0.07% and 8.13%, respectively. For instances of set B, fifteen instances over twenty-five become infeasible and ten instances lose on average the 17.74% of the revenue, where the minimum and the maximum losses of revenue are 10.08% and 24.95%, respectively. All the instances of set C become infeasible.

Note that for one instance of set A and three instances of set H, the comparison is not possible because for both models (with and without the OR-to-surgeon assignment), the integer optimization is not concluded within the given time-out and the UB resulting at the end of the B&P&Cut optimization for the model with the OR-to-surgeon assignment is greater than the best integer solution found by the B&P&Cut for the model without the OR-to-surgeon assignment, so we cannot exclude that the two models can finally converge to the same optimal integer solution.

7.3.6. Comparison with the literature

With Table 8, a comparison of the algorithms that we developed with those of [17] is proposed. We compare the average values of each instance set. Subtable *Column generation* focuses on the comparison of CG-with-Benders'-cuts with the enhanced CG with LCI cuts of [17]. For each set of instances, we reported the minimum upper bound (*Min. UB*) between the two upper bounds computed by the compared algorithms and the percentage gap of each one of the two upper bounds with the minimum; these gaps are (*gap CG-BC*) for the CG-with-Benders'-cuts and (*gap CG-LCI*) for the enhanced CG with LCI cuts. Subtables *BP Upper-Bound* and *BP Lower-Bound* focus on the comparison of the B&P&Cut that we developed with the B&P with LCI cuts of [17] (B&P&LCI). In subtable BP Upper-Bound, for each set of instances, we report the minimum UB (*Min. UB*) between

the two upper bounds computed by the compared algorithms and the percentage gap of each upper bound with the minimum; the gaps are (*gap BP-BC*) for the B&P&Cut and (*gap BP-LCI*) for the B&P&LCI of [17]. Subtable BP Lower-Bound has the same structure of subtable BP Upper-Bound, but it is for a comparison of obtained integer solutions; we report the maximum lower bound (*Max. LB*) and the percentage gap of the lower bound of each algorithm with the maximum, where gaps are (*gap BP-BC*) for the B&P&Cut and (*BP-LCI*) for the B&P&LCI of [17].

Table 8 shows that our CG-with-Benders'-cuts provides a better LP relaxation than the CG with LCI of [17]; this is true for every instance of the sets A and B, but sets A100 and A120 make an exception (column CG-BC of subtable *Column generation*). For set C, our algorithm provides the same LP relaxation. A possible explanation for the lower quality problem LP relaxation that the CG-with-Benders'-cuts delivers for sets A100 and A120 with respect to that delivered by the CG with LCI of [17] is as it follows.

The Benders-like cuts generated through the CG-with-Benders'-cuts algorithm are constraints on the aggregated selection of surgeries. Each of these cuts enforces a superior limit on the objective function value and such a superior limit is expressed as a linear combination of surgery selection variables. If the problem instance has a large number of surgeries, with many surgeries that are not mandatory, a very fractional LP solution of the RMP is more likely. The approach of [17] is quite different, the generated LCIs enforce constraints on the number of surgeries, of the same surgeon, that can be scheduled in the same OR and on the same day. It seems reasonable to argue that such a latter approach can be more effective for instances with a large number surgeries that are not mandatory and thus likely very fractional LP solutions.

The LP relaxation of the problem resulting at the end of the integer optimization of our B&P&Cut is better in most of the cases. Even if our B&P&Cut significantly improves the LP relaxation of sets A100 and A120, sets A100, A120 and B60 make an exception and the problem LP relaxation is of a slightly lesser quality than that of the B&P&LCI.

Integer solutions delivered by our B&P&Cut are frequently of lesser quality and, in some cases, they are missing, but it is relevant to point out that, in [17], the B&P&LCI is executed with a warm start taking as input a feasible integer solution computed

with a heuristic procedure, and we did not do that for executing our B&P&Cut.

If the solution computed with the heuristic that we developed is used as an initial set of columns for the CG, there is no significant benefit.

In Table 9, we report from the paper the computation time of the algorithms of [17]. The algorithms are: the CG-enhanced (the CG with dominance rules applied), the CG with LCI and the B&P with LCI; respectively, columns (*CG-Enh*), (*CG-LCI*) and (*BP-LCI*). The table reports also the computation time for our algorithms: CG, CG-with-Benders'-cuts and B&P&Cut. The times of our algorithms are scaled as if they are run on the same processor used by [17] (*i.e.*, an *Intel Xeon X5675* at 3.07 GHz).

The algorithms that we developed are faster on average than those of [17]. The difference of computation times is very large for the two CG algorithm before any cut is applied (our CG *versus* CG-Enh of [17]). The difference is reduced once the cuts are added. The computation of the LCIs of [17] appears faster than that of the Benders' cuts we developed. Comparing the computation times of CG-with-Benders'-cuts and CG-with-LCIs of [17], for small- and medium-sized instances of sets A and B, our algorithm is faster, and for large-sized instances (100 and 120 surgeries) of the same sets, computation times are comparable. Over sets C, our algorithm computes the LP relaxation of the problem in less than 1 minute, whereas the CG with LCI needs more than 1 hour. According to the numbers reported in Table 9, our B&P&Cut and that of [17] compute the UB and the integer problem solution approximately in the same time.

8. Conclusion

In this work, we defined a IORPS problem including, with fine detail, constraints that are common in practice and in the literature for similar problems and constraints for HRs other than surgeons (*i.e.*, nurses). The defined problem has sequence-dependent OR cleaning times related to consecutive surgeries with different infection levels. The integrated planning and scheduling is important because operating theaters working without detailed schedules for the all relevant resources are frequently perturbed by unexpected resource unavailability. To solve this hard problem, we have devised a branch-and-price-and-cut algorithm re-

Instance	Column generation			BP Upper-Bound			BP Lower-Bound		
	Min. UB	gap CG-BC (%)	gap CG-LCI (%)	Min. UB	gap BP-BC (%)	gap BP-LCI (%)	Max. LB	gap BP-BC (%)	gap BP-LCI (%)
A40	1407	0.00	0.28	1403	0.00	0.00	1403	0.00	0.00
A60	1964	0.00	0.56	1962	0.00	0.46	1960	0.82	0.00
A80	2198	0.00	0.32	2196	0.00	0.32	2174	1.79	0.00
A100	2231	4.03	0.00	2297	1.04	0.00	2255	2.62	0.00
A120	2218	6.58	0.00	2298	2.87	0.00	2266	2.60	0.00
B40	1365	0.00	0.15	1361	0.00	0.00	1361	0.07	0.00
B60	1859	0.00	0.05	1853	0.05	0.00	1843	0.43	0.00
B80	2143	0.00	1.40	2142	0.00	0.89	2132	0.52	0.00
B100	2389	0.00	0.54	2387	0.00	0.59	2337	3.85	0.00
B120	2399	0.00	2.42	2399	0.00	2.42	2355	4.80	0.00
C60	576	0.00	0.00	576	0.00	0.00	574	0.00	0.52
C80	792	0.00	0.00	792	0.00	0.00	713	3.79	0.00
C100	978	0.00	0.00	977	0.00	0.10	860	14.77	0.00
Average		0.82	0.44		0.30	0.37		2.77	0.04

Table 8: Comparison with results from the literature: UBs and LBs comparison

Instance	CG		CG with cuts		BP	
	CG [sec.]	CG-Enh [sec.]	CG-BC [sec.]	CG-LCI [sec.]	BP-BC [sec.]	BP-LCI [sec.]
A40	3	25	4	35	84	84
A60	20	108	40	129	8800	8659
A80	273	407	538	424	8814	8503
A100	536	1960	2143	2158	8878	9603
A120	425	1337	2750	1055	8955	8295
B40	5	35	7	42	3826	2469
B60	50	137	73	187	8802	8177
B80	150	736	467	1662	8826	9784
B100	1248	2349	4968	4633	8897	14676
B120	1328	4807	8092	5002	9125	13390
C60	32	1964	33	2329	7793	11333
C80	29	3508	29	6034	9949	15580
C100	11	4916	13	4729	8797	15545
Average	316	1715	1474	2186	7811	9700

Table 9: Comparison with results from the literature: computation times

lying on a label-correcting algorithm for solving the pricing problems and a Benders' cutting procedure for tightening the LP relaxation of the problem. The effectiveness of this solution method is demonstrated through a set of computational experiments. Our algorithm outperforms competing methods from the literature and the commercial solver (CPLEX). We hope that our devised Benders'-like cutting procedure can represent a starting point for future research of problems with several side resources to consider since the approach is sufficiently generic to be applied for an arbitrary number of resources. To this purpose, solving some technical difficulties is may be required.

Regarding the OR scheduling problem considered in this work, future research could advance in two possible directions: (i) to consider some source of uncertainty and (ii) to tackle problems that hierar-

chically follow the considered OR scheduling. Uncertain surgery durations and random resource consumptions due to add-on surgical cases that may show up after the OR scheduling can be considered as sources of uncertainty. For both cases, the optimization objective may be minimizing the average shortage of the available resource capacity or the chance of such shortages. Stochastic optimization methods are unfortunately able to solve only very small instances of IORPS problems, and an effort is required to tackle solution difficulties. Two problems that hierarchically follow the generation of OR schedules are the assignment of nurses to surgical cases and the OR rescheduling problem to include add-on urgent surgeries that cannot wait until the next complete OR scheduling.

Acknowledgments The authors would like to thank the operating theater staff of the University Hospital of Saint-Etienne from which they have been received and from which they got data and useful comments.

APPENDIX

Mixed-Integer Linear Programming Model Decision variables:

x_{ijtk} , a binary variable that takes value 1 if the surgery i is scheduled for day j and time t in OR k .

y_{kmj} , a binary variable that takes value 1 if the OR k is assigned to SG m for day j .

The OR scheduling with HR constraints MILP reads:

$$\max \sum_{i \in I} \sum_{j \in J} \sum_{t \in T_j} \sum_{k \in K} c_i x_{ijtk} \quad (52)$$

$$\sum_{j \in J} \sum_{t \in T_j} \sum_{k \in K} x_{ijtk} = 1 \quad (i \in I_{dd}) \quad (53)$$

$$\sum_{j \in J} \sum_{t \in T_j} \sum_{k \in K} x_{ijtk} \leq 1 \quad (i \in I_{nc}) \quad (54)$$

$$\sum_{i \in I_k} \sum_{t'=t-p_i+1}^t x_{ijt'k} \leq 1 \quad (t \in T, j \in J, k \in K) \quad (55)$$

$$\sum_{i \in I_a} \sum_{t'=t-p_i+1}^t \sum_{k \in K} x_{ijt'k} \leq 1 \quad (t \in T, j \in J, a \in A) \quad (56)$$

$$\sum_{t'=t+p_i}^{t+p_i+OCT(i,i')-1} \bar{c}_{ii'} x_{ijt'k} \leq 1 - x_{ijtk} \quad ((i, i') \in I : i \neq i', t \in T, j \in J, k \in K) \quad (57)$$

$$\sum_{i \in I_a} \sum_{t \in T_j^a} p_i x_{ijt} \leq T_{aj} \quad (j \in J, a \in A) \quad (58)$$

$$\sum_{t \in T_j} x_{ijtk} \leq y_{kmj} \quad (i \in I_m, j \in J, m \in SG, k \in K) \quad (59)$$

$$\sum_{m \in SG} y_{kmj} \leq |K_j| \quad (j \in J, k \in K) \quad (60)$$

$$\sum_{i \in I_m} \sum_{t'=t-p_i+1}^t n_i x_{ijt'k} \leq |B_{mj}| \quad (t \in T_j, j \in J, k \in K) \quad (61)$$

$$x_{ijtk} = 0 : j > dd_i \vee j < rd_i \quad (i \in I, t \in T, j \in J; k \in K) \quad (62)$$

$$x_{ijtk} \in \{0, 1\} \quad (i \in I, t \in T, j \in J; k \in K) \quad (63)$$

$$y_{kmj} \in \{0, 1\} \quad (k \in K, m \in SG, j \in J) \quad (64)$$

The objective function maximizes the total revenue of surgeries scheduled in the planning horizon. Constraints (53) enforce that mandatory surgeries are scheduled only once in the planning horizon. Constraints (54) enforce not mandatory surgeries to be scheduled at most once in the planning horizon. Constraints (55) enforce that surgeries do not overlap in the same OR. Constraints (56) enforce that surgeries of the same surgeon do not overlap in the schedule. Constraints (57) enforce that an OCT is scheduled between any pair of consecutive surgeries scheduled for the same OR and day in the planning horizon if an OCT is required in between. Constraints (58) enforce that the daily surgeon maximum surgery time is not exceeded in the schedule for every surgeon and day of the planning horizon. Constraints (59) enforce that only surgeries belonging to same SG can be scheduled for the same OR and day in the planning horizon. Constraints (60) enforce that only one SG can be assigned to an OR for a every day in the planning horizon. Constraints (61) enforce that surgeries scheduled do not require more nurses than those available; this is for every SG, day and time in the planning horizon. Constraints (62) enforce that any surgery is scheduled not before its release date and not after its due date. With (63) and (64) variable x_{ijtk} and y_m domains are defined.

References

- [1] A. Atamtürk. Cover and pack inequalities for (mixed) integer programming. *Annals of Operations Research*, 139(1):21–38, 2005.

- [2] S. Batun, B. T. Denton, T. R. Huschka, and A. J. Schaefer. Operating room pooling and parallel surgery processing under uncertainty. *INFORMS journal on Computing*, 23(2):220–237, 2011.
- [3] B. Cardoen, E. Demeulemeester, and J. Beliën. Sequencing surgical cases in a day-care environment: an exact branch-and-price approach. *Computers & Operations Research*, 36(9):2660–2669, 2009.
- [4] B. Cardoen, E. Demeulemeester, and J. Beliën. Operating room planning and scheduling: A literature review. *European journal of operational research*, 201(3):921–932, 2010.
- [5] P. M. Castro and I. Marques. Operating room scheduling with generalized disjunctive programming. *Computers & Operations Research*, 64:262–273, 2015.
- [6] P. M. Castro, L. J. Zeballos, and C. A. Méndez. Hybrid time slots sequencing model for a class of scheduling problems. *AIChE Journal*, 58(3):789–800, 2012.
- [7] J.-F. Cordeau, F. Soumis, and J. Desrosiers. Simultaneous assignment of locomotives and cars to passenger trains. *Operations research*, 49(4):531–548, 2001.
- [8] J.-F. Cordeau, G. Stojković, F. Soumis, and J. Desrosiers. Benders decomposition for simultaneous aircraft routing and crew scheduling. *Transportation science*, 35(4):375–388, 2001.
- [9] K. Fang, N. A. Uhan, F. Zhao, and J. W. Sutherland. Scheduling on a single machine under time-of-use electricity tariffs. *Annals of Operations Research*, 238(1-2):199–227, 2016.
- [10] H. Fei, C. Chu, and N. Meskens. 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, 2009.
- [11] H. Fei, N. Meskens, and C. Chu. A planning and scheduling problem for an operating theatre using an open scheduling strategy. *Computers & Industrial Engineering*, 58(2):221–230, 2010.
- [12] D. Feillet, P. Dejax, M. Gendreau, and C. Gueguen. An exact algorithm for the elementary shortest path problem with resource constraints: Application to some vehicle routing problems. *Networks: An International Journal*, 44(3):216–229, 2004.
- [13] A. Fréville and S. Hanafi. The multidimensional 0-1 knapsack problem? bounds and computational aspects. *Annals of Operations Research*, 139(1):195, 2005.
- [14] T. Garaix, C. Artigues, D. Feillet, and D. Josselin. Optimization of occupancy rate in dial-a-ride problems via linear fractional column generation. *Computers & Operations Research*, 38(10):1435–1442, 2011.
- [15] Z. Gu, G. L. Nemhauser, and M. W. Savelsbergh. Lifted cover inequalities for 0-1 integer programs: Computation. *INFORMS Journal on Computing*, 10(4):427–437, 1998.
- [16] A. Guinet and S. Chaabane. Operating theatre planning. *International Journal of Production Economics*, 85(1):69–81, 2003.
- [17] S. H. Hashemi Doulabi, L.-M. Rousseau, and G. Pesant. A constraint-programming-based branch-and-price-and-cut approach for operating room planning and scheduling. *INFORMS Journal on Computing*, 28(3):432–448, 2016.
- [18] J. N. Hooker and G. Ottosson. Logic-based benders decomposition. *Mathematical Programming*, 96(1):33–60, 2003.
- [19] R. Ibrahim and S.-H. Kim. Is expert input valuable? the case of predicting surgery duration. *Seoul Journal of Business*, Forthcoming, 2019.
- [20] A. Jebali, A. B. H. Alouane, and P. Ladet. Operating rooms scheduling. *International Journal of Production Economics*, 99(1-2):52–62, 2006.
- [21] L. Kowalik and J. Lauri. On finding rainbow and colorful paths. *Theoretical Computer Science*, 628:110–114, 2016.
- [22] I. Marques and M. E. Captivo. Different stakeholders’ perspectives for a surgical case assignment problem: Deterministic and robust approaches. *European Journal of Operational Research*, 261(1):260–278, 2017.
- [23] I. Marques, M. E. Captivo, and M. V. Pato. An integer programming approach to elective surgery scheduling. *OR spectrum*, 34(2):407–427, 2012.
- [24] I. Marques, M. E. Captivo, and M. V. Pato. Scheduling elective surgeries in a portuguese hospital using a genetic heuristic. *Operations Research for Health Care*, 3(2):59–72, 2014.
- [25] A. Mercier, J.-F. Cordeau, and F. Soumis. A computational study of benders decomposition for the integrated aircraft routing and crew scheduling problem. *Computers & Operations Research*, 32(6):1451–1476, 2005.
- [26] A. Mercier and F. Soumis. An integrated aircraft routing, crew scheduling and flight retiming model. *Computers & Operations Research*, 34(8):2251–2265, 2007.
- [27] B. Roland, C. Di Martinelly, F. Riane, and Y. Pochet. Scheduling an operating theatre under human resource constraints. *Computers & Industrial Engineering*, 58(2):212–220, 2010.
- [28] V. Roshanaei, K. E. Booth, D. M. Aleman, D. R. Urbach, and J. C. Beck. Branch-and-check methods for multi-level operating room planning and scheduling. *International Journal of Production Economics*, 220:107433, 2020.
- [29] V. Roshanaei, C. Luong, D. M. Aleman, and D. R. Urbach. Collaborative operating room planning and scheduling. *INFORMS Journal on Computing*, 29(3):558–580, 2017.
- [30] T. A. Silva, M. C. de Souza, R. R. Saldanha, and E. K. Burke. Surgical scheduling with simultaneous employment of specialised human resources. *European Journal of Operational Research*, 245(3):719–730, 2015.
- [31] J. Van den Akker, C. A. Hurkens, and M. W. Savelsbergh. Time-indexed formulations for machine scheduling problems: Column generation. *INFORMS Journal on Computing*, 12(2):111–124, 2000.
- [32] J. T. van Essen, E. W. Hans, J. L. Hurink, and A. Oversberg. Minimizing the waiting time for emergency surgery. *Operations Research for Health Care*, 1(2-3):34–44, 2012.
- [33] B. Vijayakumar, P. J. Parikh, R. Scott, A. Barnes, and J. Gallimore. A dual bin-packing approach to scheduling surgical cases at a publicly-funded hospital. *European Journal of Operational Research*, 224(3):583–591, 2013.