Journal of Scheduling (2020) 23:265–288
https://doi.org/10.1007/s10951-020-00639-6

A diving heuristic for planning and scheduling surgical cases in the operating room department with nurse re-rostering

Babak Akbarzadeh[1] · Ghasem Moslehi[2] · Mohammad Reisi-Nafchi[2] · Broos Maenhout[1]

Published online: 28 January 2020

Abstract

The decisions in the operating room scheduling process related to the case mix planning, the master surgery schedule and the nurse roster are based on the expected demand, predicted by historical data. Patients are only scheduled in the operational phase when the actual demand is known. However, the actual patient demand may differ from the expected demand. In this paper, we integrate the surgical case planning and scheduling problem and include the nurse re-rostering decision and nurse assignment to specific patients in order to utilise the operating room department as efficiently as possible and maximise the operating room profit. We propose a two-phase heuristic that uses the LP solution generated via column generation to construct a high-quality feasible solution. Computational experiments have been conducted on a diverse artificial data set generated in a controlled and structured manner and real-life data from the Sina Hospital (Tehran, Iran). We show that the presented approach is able to produce (near-)optimal solutions and benchmark the procedure with other optimisation strategies and solution methodologies.

Keywords Surgical case planning and scheduling · Nurse re-rostering · Nurse patient assignment · Column generation · Diving heuristic

1 Introduction

The operating room (OR) department is key in the operations of hospitals as more than 60% of hospital admissions belong to this department accounting for about 40% of the hospital costs (Pham and Klinkert 2008; Villarreal and Keskinocak 2016). In order to maximise the utilisation of the OR resources, the OR manager turns to the OR scheduling process that consists in most hospitals of three hierarchical decision phases (Batun et al. 2011; Fügener et al. 2014). The first phase concerns the distribution of the OR time between the medical disciplines or surgeons based on the expected patient demand, which is known as the case mix planning problem. The second phase encompasses the master surgery scheduling (MSS) problem and assigns a block of consecutive time slots in a specific operating room on a particular day to a surgeon based upon the case mix planning. The duty roster of the nurses working in the OR department, called surgical nurses, is constructed based on this master surgery schedule, which takes thus the expected patient demand into account (Beliën and Demeulemeester 2008). According to Guo et al. (2014), the cost of the nursing resources in the OR department is no less than 37%. The third phase is related to patient scheduling and is typically decomposed in two steps. First, surgical cases are assigned to a specific day in the time horizon (i.e. the surgical case planning (SCP) problem) and then the planned surgeries are sequenced and assigned to a specific time slot on that day (i.e. the Surgical Case Scheduling (SCS) problem). The hierarchical structure implies that the scheduled patients are assigned to a specific day and time slot taking the availability of resources into account (surgeon, OR and nurses).

✉ Broos Maenhout
Broos.Maenhout@Ugent.be

Babak Akbarzadeh
Babak.Akbarzadeh@ugent.be

Ghasem Moslehi
Moslehi@cc.iut.ac.ir

Mohammad Reisi-Nafchi
Reisi.M@cc.iut.ac.ir

[1] Faculty of Economics and Business Administration, Ghent University, Tweekerkenstraat 2, 9000 Ghent, Belgium

[2] Department of Industrial and Systems Engineering, Isfahan University of Technology, Isfahan 84156-83111, Iran

In order to maximise the utilisation of the OR department, we integrate in this paper surgical case planning and surgical case scheduling given a master surgery schedule to address the surgical case planning and scheduling (SCPS) problem. The problem arises at the operational level of decision making where the surgery demand for all disciplines is accurate. The surgical cases are assigned to particular time slots and rooms for the upcoming week respecting the master surgery schedule such that the OR profit is maximised. Note that we consider only elective surgical cases since in practice some slack capacity is commonly reserved for emergency cases by the OR manager. In addition, in order to further improve the efficiency of the OR department, we incorporate the re-rostering of nursing resources such that the number of nurse shift duties over the planing horizon is minimised. The required adaptations to the nurse rosters are motivated by the fact that the previously announced roster is designed based upon the master surgery schedule (Beliën and Demeulemeester 2008). The actual demand for a particular surgeon or discipline, however, may differ from the expected demand such that changes in the nurse roster are necessary to match supply and demand for surgical nurses. In this way, nurses may be (re-)assigned to additional or other nurse shift duties or their shift duties may be cancelled in order to optimise profit. Note that we want to keep the number of changes as small as possible since only assignments are possible conform to their original roster or the nurse preferences. In this operational phase, surgical nurses are assigned to particular shift duties and specific surgical cases in accordance with the patient's needs. In this way, the minimum required number of nurse duties can be determined in a more accurate manner taking the differences in nurse requirements per surgery into account.

The problem under study is formulated as a decomposed mixed integer programming (MIP) problem thriving on two types of column variables, i.e. (1) feasible patient schedules for a single day and (2) the nurse duty shift assignment for a particular nurse, day and shift characterised by specific nurse–patient assignments. We propose a two-step heuristic solution methodology. In the first step, we apply a column generation algorithm and include different dedicated speed-up mechanisms to solve the relaxed linear programming (LP) problem. In the second step, a greedy diving heuristic is used to drive the optimal LP solution to integrality. The proposed methodology leads to optimal or near-optimal solutions in a short time span. In the computational experiments, we validate the algorithm design choices and benchmark the procedure with other solution methodologies. The experiments are conducted on an artificial data set generated in a controlled and structured manner and on real-life data from the Sina Hospital (Tehran, Iran).

The remainder of the paper is organised as follows. In Sect. 2, we discuss the related literature. In Sect. 3, we give a detailed description and a mathematical formulation of the problem under study. Section 4 discusses the two-phase solution methodology. In Sect. 5, the computational experiments are presented. In Sect. 6, we validate the proposed methodology and show its benefits for a real-life case study. Section 7 provides concluding remarks and directions for future research.

2 Literature review

In this section, we review the related literature for the problem under study. Table 1 provides a synthesis of the relevant literature, which is organised according to the problem definition and is further discussed below. The literature on the surgical case planning and scheduling problem is discussed in Sect. 2.1, and the literature that considers surgical nurse resources in the OR department is reviewed in Sect. 2.2. The contribution of this paper is discussed in Sect. 2.3.

2.1 Surgical case planning and scheduling

There exists a vast amount of literature that concerns the planning and scheduling of patients in the OR department. The topic has been investigated widely, and some recent literature reviews are available, i.e. (Cardoen et al. 2010; Guerriero and Guido 2011; Van Riet and Demeulemeester 2015; Gür 2018). In the following, we discuss the relevant problem context and definition, the objectives and the relevant solution methodologies for surgical case planning and scheduling.

Problem context and definition

Surgical case planning and scheduling is organised either according to a block or open booking system. In the block booking system, a master surgery schedule divides the OR time between different (groups of) surgeons into blocks based on the case mix planning. The surgeons plan their surgical cases within their assigned time slots (Guerriero and Guido 2011). In the open scheduling strategy, no blocks are assigned to the surgeons, which submit their surgical cases for the next day(s) until their maximum number of allowed cases is reached or until the OR capacity is filled (Magerlein and Martin 1978). The precise surgery dates and start times are usually determined via an online algorithm based on the considerations of different stakeholders (e.g. surgeon, OR resources, patient) (Zhang and Xie 2015).

Surgical case planning and scheduling is situated on the operational planning level and is typically comprised of two steps. In the surgical case planning, the surgical cases are assigned to the available time blocks for a specific surgeon coming from the master surgery schedule. This entails that a surgery date and operating room is assigned to each surgical case (Lamiri et al. 2008; Fei et al. 2009a; Agnetis et al. 2014). In the second step, the patient schedule or sequence of

Table 1 Synthesis of the related literature

Related studies	Problem definition				Objective function				
	Surgical Case planning	Surgical Case scheduling	Surgical Case planning and scheduling	Surgical Nurse resources	Productivity related	Resource related	Time related	Preference related	Cost related
Lamiri et al. (2008)	✓								✓
Fei et al. (2009a)	✓						✓		✓
Fei et al. (2009b)	✓						✓		✓
Wang et al. (2014)	✓								✓
Agnetis et al. (2014)	✓							✓	
Cardoen et al. (2009)		✓				✓		✓	
van Essen et al. (2012)		✓						✓	
Aringhieri et al. (2015)		✓						✓	
Ozkarahan (1995)	✓	✓					✓		
Guinet and Chaabane (2003)	✓	✓					✓		
Hadj et al. (2006)	✓	✓							✓
Fei et al. (2010)	✓	✓					✓		✓
Souki and Rebai (2012)	✓	✓						✓	
Marques et al. (2014)			✓		✓		✓		
Doulabi et al. (2016)			✓				✓		
Mobasher et al. (2011)				✓			✓	✓	
Lim et al. (2016)				✓		✓	✓	✓	
Di Martinelly and Meskens (2017)				✓				✓	
Pham and Klinkert (2008)		✓		✓			✓		
Meskens et al. (2013)		✓		✓			✓	✓	
Xiang et al. (2014)		✓		✓			✓		
Monteiro et al. (2015)		✓		✓			✓	✓	
Silva et al. (2015)		✓		✓		✓			
Latorre-Núñez et al. (2016)		✓		✓			✓		
Ogulata and Erol (2003)	✓	✓		✓		✓		✓	
Chaabane et al. (2008)	✓	✓		✓					✓
Vijayakumar et al. (2013)	✓	✓		✓				✓	
Roland et al. (2007)			✓	✓					✓
Roland et al. (2010)			✓	✓					✓
Roland and Riane (2011)			✓	✓				✓	✓
Di Martinelly et al. (2014)			✓	✓					✓
Riise et al. (2016)			✓	✓		✓	✓		✓
This paper			✓	✓				✓	✓

Table 1 continued

Related studies	Surgical nurse				Solution method		
	Nurse constraint	Nurse assignment	Nurse shift staffing	Nurse re-rostering	Heuristic procedure	Exact procedure	Column generation
Lamiri et al. (2008)					✓		✓
Fei et al. (2009a)					✓		✓
Fei et al. (2009b)						✓	✓
Wang et al. (2014)					✓		✓
Agnetis et al. (2014)					✓		
Cardoen et al. (2009)						✓	✓
van Essen et al. (2012)					✓		
Aringhieri et al. (2015)					✓		
Ozkarahan (1995)					✓		
Guinet and Chaabane (2003)					✓		
Hadj et al. (2006)					✓		
Fei et al. (2010)					✓		
Souki and Rebai (2012)					✓		
Marques et al. (2014)					✓		
Doulabi et al. (2016)						✓	✓
Mobasher et al. (2011)	✓	✓			✓		
Lim et al. (2016)	✓	✓			✓		
Di Martinelly and Meskens (2017)	✓	✓				✓	
Pham and Klinkert (2008)	✓	✓				✓	
Meskens et al. (2013)	✓	✓				✓	
Xiang et al. (2014)	✓	✓			✓		
Monteiro et al. (2015)	✓	✓				✓	
Silva et al. (2015)	✓				✓		
Latorre-Núñez et al. (2016)	✓				✓		
Ogulata and Erol (2003)	✓	✓			✓		
Chaabane et al. (2008)	✓				✓		
Vijayakumar et al. (2013)	✓				✓		
Roland et al. (2007)	✓				✓		
Roland et al. (2010)	✓				✓		
Roland and Riane (2011)	✓				✓		
Di Martinelly et al. (2014)	✓		✓			✓	
Riise et al. (2016)	✓	✓			✓		
This paper	✓	✓	✓	✓	✓	✓	✓

the planned surgical cases is determined as precise surgery start times are specified (Pham and Klinkert 2008; Cardoen et al. 2009; Essen et al. 2012; Monteiro et al. 2015). This step considers the assignment of patients to resources, i.e. physical resources (such as operating rooms, equipment, post-anaesthesia care unit recovery beds) and medical staff (such as surgeons, nurses, anaesthetists) (Meskens et al. 2013; Di Martinelly et al. 2014; Latorre-Núñez et al. 2016). Decisions taken in the planning step will obviously affect the scheduling decision and the resulting solution quality. Therefore, a significant number of authors have tried to tackle this problem using a two-step hierarchical method, which considers the planning and the scheduling problem as two different but related sequential subproblems (Ozkarahan 1995; Guinet and Chaabane 2003; Fei et al. 2010; Essen et al. 2012; Souki and Rebai 2012; Vijayakumar et al. 2013; Aringhieri et al. 2015). Recently, different authors tackled the planning and scheduling problem in an integrated manner in which all decisions related to the date and the start time are made together (Roland et al. 2010; Di Martinelly et al. 2014; Doulabi et al. 2016; Di Martinelly and Meskens 2017). Research on this topic is limited since tackling the problem in an integrated way increases the complexity of synchronisation and thereby the computational effort.

Objectives

SCPS is complex since the objectives of all the stakeholders need to be considered. A surgical plan can be evaluated by objectives related to productivity or patient through (e.g. Marques et al. 2014), related to the resources, i.e. resource cost, utilisation or the levelling of resources (e.g. Riise et al. 2016), related to the time, i.e. overtime and makespan (e.g. Ozkarahan 1995; Fei et al. 2009a), related to the preferences of surgeons, patient priorities and waiting time of surgical cases (e.g. Pham and Klinkert 2008; Essen et al. 2012; Roland and Riane 2011; Souki and Rebai 2012; Di Martinelly et al. 2014; Monteiro et al. 2015; Riise et al. 2016) and related to the cost of the performed surgeries and closely connected the revenues (Lamiri et al. 2008; Fei et al. 2009a). As a result, different papers study a multi-objective optimisation problem. Cardoen et al. (2009) considered multiple objectives related to waiting time and priority of surgical cases, resource cost and resource levelling. Monteiro et al. (2015) proposed a multi-objective model to minimise the makespan and the overtime resource cost. Meskens et al. (2013) and Di Martinelly and Meskens (2017) considered overtime and surgical team preferences. Different papers (e.g. Hadj et al. 2006; Lamiri et al. 2008; Roland et al. 2010; Latorre-Núñez et al. 2016) consider different types of objectives and formulate the objective function on a uniform basis in terms of costs. As a result, minimising costs and, closely related, maximising revenues are widely studied objectives in OR scheduling. Doulabi et al. (2016) and Marques et al. (2014) optimise their revenues by avoiding unused capacity as they maximise the total duration or the total number of the scheduled surgical cases. Roland et al. (2007) minimise the opening costs of the operating rooms and the overtime. Costs can be further reduced by considering human resource capacity and optimising the resource cost (Roland et al. 2010; Velasquez and Melo 2006).

Solution methodology

Solving the SCPS problem to optimality in a real-life environment is very difficult due to the long list of surgical cases and the specific combination of objective(s) and constraints (Marques et al. 2014). The formulations for the SCPS problem involve a huge number of binary variables that increases exponentially with the problem size and significantly impacts the problem complexity. Moreover, large-size problem instances require not only additional computational effort but also some technical effort (e.g. the development of a new model). In the literature, both exact and heuristic methodologies are devised. We refer to the overview of Samudra et al. (2016) for a comprehensive review of the appropriate solution methodologies.

Different heuristic methodologies are developed based upon the hierarchical structure of the problem and solve the problem in two sequential steps, i.e. first they solve the SCP problem, and the obtained surgical case plan is fixed and used as input for the SCS problem (cf. supra). Vijayakumar et al. (2013) compare an integrated methodology, which embodies the simultaneous surgical case planning and scheduling, with a hierarchical solution approach and demonstrate that even for small-sized instances the solutions generated by the hierarchical approach lead to more unscheduled surgical cases due to the difference between objectives in the two aforementioned problems. The integrated approach leads to a more detailed planning problem and increases the chance of obtaining a stable schedule that can be successfully implemented (Doulabi et al. 2016).

Different exact and heuristic methodologies, relevant for the SCPS problem, thrive on mathematical programming and a decomposed problem formulation (Lamiri et al. 2008; Fei et al. 2010; Cardoen et al. 2009; Wang et al. 2014; Doulabi et al. 2016). These studies typically reformulate the original problem by applying the Dantzig–Wolfe decomposition and use column generation to find the optimal linear programming (LP) solution, based upon which a high-quality integer solution is found. The defined column variables in the literature lead to a different problem formulation to solve the SCPS problem. Fei et al. (2010), Lamiri et al. (2008) and Wang et al. (2014) define a column as a feasible plan assigning surgical cases to one operating room on a single day. Fei et al. (2010) utilise the obtained surgical case plan as input to schedule the surgical cases via a genetic algorithm. Lamiri et al. (2008) use a three-phase heuristic solution methodology. They first apply column generation to solve the relaxed LP model. An integer solution is obtained by retaining all columns with a positive integer value, assigning surgical cases to specific operating rooms. The remainder of the surgical cases are assigned heuristically, i.e. one-by-one to the operating rooms based on their costs. The feasible solution is further improved by adding neighbourhood search methods such as assigning surgical cases to other operating rooms or the pair-wise exchange of cases. Fei et al. (2009b) and Wang et al. (2014) apply column generation to solve the LP-relaxed problem and a diving heuristic to find a feasible integer solution. The diving heuristic repetitively fixes particular columns and uses column generation in each iteration until an integer solution is found. Cardoen et al. (2009) model a sequenced group of surgical cases for one specific day and surgeon as a column. Doulabi et al. (2016) define a column as a schedule for a set of surgeries with fixed start times in a single operating room on a given day. In their solution methodology, they first apply column generation to solve the LP relaxation of the reformulated model and either heuristic principles or an exact approach is used to transform the LP optimal solution into an integer solution. Their exact approach comprises a branch-and-price with different speed-up techniques to accelerate the branching procedure. They indicated that closing the optimality gap for real-life problems is time-consuming.

2.2 Nurse scheduling in the operating department

Operating room departments, and hospitals in general, suffer from a shortage of nurses and overtime hours are often inevitable. The OR manager's role is to coordinate the surgical teams in order to perform surgical operations efficiently and safely on time (Di Martinelly and Meskens 2017). Different papers related to surgical case planning and scheduling consider the presence of nursing resources (e.g. Ogulata and Erol 2003; Chaabane et al. 2008; Meskens et al. 2013; Xiang et al. 2014). In its simplest form, only the maximum number of surgical nurses is considered as a resource constraint in order to construct a feasible surgical case plan and/or schedule (e.g. Silva et al. 2015; Latorre-Núñez et al. 2016). Ben-Arieh et al. (2014) and Wang et al. (2015) propose a mathematical model to determine a surgery date for surgical cases (i.e. surgical case planning) and the required number of surgical nurses for each shift, i.e. the nurse shift staffing problem. Di Martinelly et al. (2014) investigated surgical case planning and scheduling and calculated the minimum required number of nurses for each shift. They considered six different scenarios to deal with trade-offs between the cost of opening operating rooms, the cost associated with nurse salaries and overtime payments.

In order to meet the patient's needs, some studies assign surgical nurses directly to specific surgical cases in the operational phase in accordance to their qualifications and competences, which is referred to as the nurse surgical assignment problem. There are different constraints postulated in the literature related to the surgical nurse assignment problem that resemble real-world situations. Most of these constraints model the availability and skills of the nursing resources (Ogulata and Erol 2003; Chaabane et al. 2008; Meskens et al. 2013; Xiang et al. 2014). Pham and Klinkert (2008) relate the assignment of specific nursing teams to particular surgical cases. A few studies consider multiple types of personnel resources (e.g. technicians, nurses) and build heterogeneous surgical teams taking their availabilities into account (Roland et al. 2010; Roland and Riane 2011; Silva et al. 2015; Monteiro et al. 2015). Meskens et al. (2013) and Di Martinelly and Meskens (2017) construct these surgical teams by additionally considering the team preferences. Note that in the literature the surgical nurse assignment is typically carried out in an additional optimisation step after the surgical case schedule has been composed when the start times of operations are known (Mobasher et al. 2011; Lim et al. 2016; Di Martinelly and Meskens 2017).

2.3 Contribution of this paper

The contribution of this paper is twofold. First, we consider the surgical case planning and scheduling problem, the nurse re-rostering and nurse assignment problem. To the best of our knowledge, these problems have not been investigated in an integrated model and we discuss the value of integration of the nurse re-rostering and nurse assignment to surgical cases in the problem definition. Second, we present a heuristic solution methodology building upon the optimal LP solution obtained via a column generation algorithm and a diving heuristic to drive the optimal LP solution to integrality. In the computational experiments, we prove that this methodology leads to (near-)optimal solutions and outperforms different alternative solution methodologies.

3 Problem description and formulation

3.1 Problem description

For the problem under study, we assume that different decisions related to the case mix planning and the master surgery schedule are taken and that the set of elective patients for the relevant planning horizon, i.e. the next upcoming days or week, is defined. The master surgery schedule and the associated nurse roster are created based on the expected patient demand. In the operational phase, the OR manager has an accurate knowledge of the actual patient demand for the upcoming days or week based upon the planned surgical cases for all surgeons and medical disciplines. Due to the operational variability, the actual patient demand can be higher or lower than the expected demand. In order to resolve this issue, the OR manager has to plan and schedule the defined set of surgical cases as efficiently as possible in order to maximise the utilisation of the resources (OR time, nurses) considering the following decisions, i.e.

1. The Surgical Case Planning and Scheduling problem (SCPS)
   The surgical cases are planned over the planning horizon and sequenced by assigning these cases to particular time slots. The operating rooms are available each day for a set of time slots. Each surgery is characterised by a deterministic surgery duration and a particular surgeon. A surgical case needs to be carried out during the blocks assigned to the particular surgeon stipulated by the master surgery schedule. The blocks in the master surgery schedule are specified by the operating surgeon, the operating room, the day and a set of time slots. Apart from a surgeon, a surgery requires a specific number of nurses, which can be dependent from surgery to surgery. All surgeries require exactly the same type of nurse competences since only the surgical cases of a medical cluster are considered containing different surgeons from the same or related disciplines. Note that the surgical team assigned to a surgery is fixed and cannot be modified during the surgery.
2. The Nurse Re-Rostering and Surgical Assignment problem (NRRSA)
   In order to facilitate the scheduling of surgical cases to time slots, nurses are assigned to specific duty shifts and surgical cases in order to ensure the required number of nursing resources are present to operate the individual surgical cases properly. The duty shifts of nurses are rescheduled such that the nurse schedule matches the actual patient demand. This implies that the set of nurses should be re-rostered for the planning horizon, since their original roster was based on the expected demand. In this decision, a nurse will be assigned to shift duties either based on the original roster or will be re-assigned to another shift based on the nurse preferences. In this way, the number of shift changes is kept to a minimum and the problem can be labelled as a minimal perturbation problem and as a nurse re-rostering problem in particular (Moz and Pato 2003; Maenhout and Vanhoucke 2011). The nurse re-rostering problem incorporates the changes in the patient demand along with the original nurse roster in the problem definition for which the solution should be as close as possible to the original roster. Nurses are additionally assigned to specific surgical cases in order to accurately determine the nurse requirements for each shift. The objective of the nurse re-rostering and surgical assignment problem is to minimise the number of shift duties. We further assume that the nursing resources are homogeneous, i.e. they have all exactly the same skills. The shift duties are organised as a set of non-overlapping, successive shifts with a fixed start and end time.

3.2 Problem formulation

In the following, we present a mathematical model for the SCPS-NRRSA problem. Note that the formulated model is based on the Dantzig–Wolfe decomposition (Breu et al. 2008), where column-based variables are included instead of the original decision variables. In the model formulation, we make use of the following two types of columns, i.e.

- A *surgical case schedule* (SCS) represents the assignment of surgical cases to time slots and the different operating rooms on a particular day.
- A *nurse duty assignment* (NDA) indicates the assigned surgical cases and shift duty on a particular day and shift for a nurse.

The problem is formulated as follows:

List of symbols
Sets

I	The set of surgeries (index i and $\bar{i}$)
N	The set of nurses (index n)
D	The set of days in the planning horizon (index d)
V	The set of shifts (index v)
T	The set of time slots per day (index $t, \bar{t}$)
R	The set of operating rooms (index r)
S	The set of surgeons (index s)
φ_s	The set of surgical cases related to surgeon s
J_d	The set of feasible surgical case schedules (SCS columns) for day d (index j)
Q_{ndv}	The set of feasible nurse duty assignments (NDA columns) for nurse n, day d and shift v (index q)

Parameters

nu_i	Required number of nurses to perform surgery i
Rev_i	Revenue for performing surgical case i
c_r^{OR}	The cost per day for opening operating room r
c_v^N	A cost for assigning a nurse to shift duty v
MA	The maximum number of shifts that can be assigned to a nurse
BM	A large positive constant
π_τ^μ	Dual price for τ^{th} constraint of type μ (indicated next to the constraints)

Column parameters

$\bar{a}_{idj}$ 1, if surgical case i is included in the SCS j at day d; 0, otherwise

$\bar{a}v_{idvj}$ 1, if surgical case i is performed in shift v in the SCS j at day d; 0, otherwise

$\bar{e}_{rdj}$ 1, if the SCS j, day d involves OR r; 0, otherwise

$\bar{P}_{i\bar{i}dj}$ 1, if surgical case i and $\bar{i}$ have time overlap in the SCS j, day d; 0, otherwise

$\bar{h}_{indvq}$ 1, if surgical case i is included in the NDA q for nurse n, day d, shift v; 0, otherwise

Decision variables

x_{dj} 1, if the surgical case schedule j is selected for day d; 0, otherwise

y_{ndvq} 1, if the nurse duty assignment q is selected for nurse n in shift v at day d; 0, otherwise

Master problem formulation

$$\text{Max } Z = \sum_{idj} Rev_i \times \bar{a}_{idj} \times x_{dj} - \sum_{rdj} c_r^{OR} \times \bar{e}_{idj} \times x_{dj} - \sum_{ndvq} c_v^N \times y_{ndvq} \tag{1}$$

The objective function (1) maximises the revenues resulting from the surgical cases performed and minimises the costs related to the total number of operating rooms used and the total number of shift duties assigned to the nurses.

$$\sum_{dj} \bar{a}_{idj} \times x_{dj} \leq 1 \qquad \forall i \qquad \pi_i^2 \tag{2}$$

$$\sum_{j} x_{dj} \leq 1 \qquad \forall d \qquad \pi_d^3 \tag{3}$$

$$\sum_{dvq} y_{ndvq} \leq MA \qquad \forall n \qquad \pi_n^4 \tag{4}$$

$$\sum_{vq} y_{ndvq} \leq 1 \qquad \forall n, d \qquad \pi_{nd}^5. \tag{5}$$

Constraint (2) stipulates that a single surgery can be carried out at most once over the planning horizon. Constraint (3) ensures that for each day at most one surgical case schedule must be selected. Constraint (4) imposes that a nurse cannot be assigned to more than *MA* shift duties during the planning horizon. Constraint (5) stipulates that a nurse cannot be assigned to more than one shift per day.

$$\sum_{j} nu_i \times \bar{a}v_{idvj} \times x_{dj} = \sum_{nq} \bar{h}_{indvq} \times y_{ndvq} \qquad \forall i, v, d \qquad \pi_{idv}^6. \tag{6}$$

Constraint (6) imposes that the required number of nurses are assigned to each surgical case and links the SCPS problem and the NRRSA problem. The left-hand side of this constraint indicates that the nurse requirements for shift v and day d are generated by the surgical cases included in the selected surgical case schedule. The right-hand side shows the assigned number of nurses as a result from the selected nurse duty assignments.

$$\sum_{ndvq} \bar{h}_{indvq} \times \bar{h}_{\bar{i}ndvq} \times y_{ndvq} \leq BM \times (1 - \sum_{j} \bar{P}_{i\bar{i}dj} \times x_{dj}) \qquad \forall i < \bar{i}, d \qquad \pi_{i\bar{i}d}^7 \tag{7}$$

$$x_{dj}, \in \{0, 1\} \qquad \forall d, j$$
$$y_{ndvq} \in \{0, 1\} \qquad \forall q, n, v, d. \tag{8}$$

Constraint (7) implies that any pair of surgical cases $(i, \bar{i})$ with an overlap in the surgical case schedule j, day d (i.e. $\bar{P}_{i\bar{i}dj} = 1$) cannot not be both included in the nurse duty assignment of a single nurse. In case the surgical cases do not overlap (i.e. $\bar{P}_{i\bar{i}dj} = 0$), a suitable upper bound value BM is equal to $nu_i + nu_{\bar{i}}$, i.e. the maximum number of nurse duty assignments that can include surgical case i and $\bar{i}$. Constraint (8) represents the binary conditions.

4 Solution methodology

In this section, we discuss a column generation-based diving heuristic solution procedure (CGD) to solve the integrated SCPS-NRRSA problem. The procedure finds first the optimal LP solution using column generation. If the LP solution is fractional, a diving heuristic is invoked to transform the LP solution into an integer solution. Figure 1 provides an overview of these two steps of the solution procedure.

The *first step* embodies a column generation procedure (CG), discussed in detail in Sect. 4.1, since it is computationally infeasible to generate all possible surgical case schedules and nurse duty assignments to solve the proposed model directly (Breu et al. 2008). The column generation procedure relaxes the integrality conditions and iterates between the restricted master problem and the two pricing problems, i.e. the surgical case schedule pricing problem and the nurse duty assignment pricing problem, to find the optimal linear programming (LP) solution. The restricted master problem relaxes the integrality conditions and works only with a restricted subset of all possible SCS and NDA column variables. Additional variables are included via the pricing step based on the dual prices returned by the restricted master

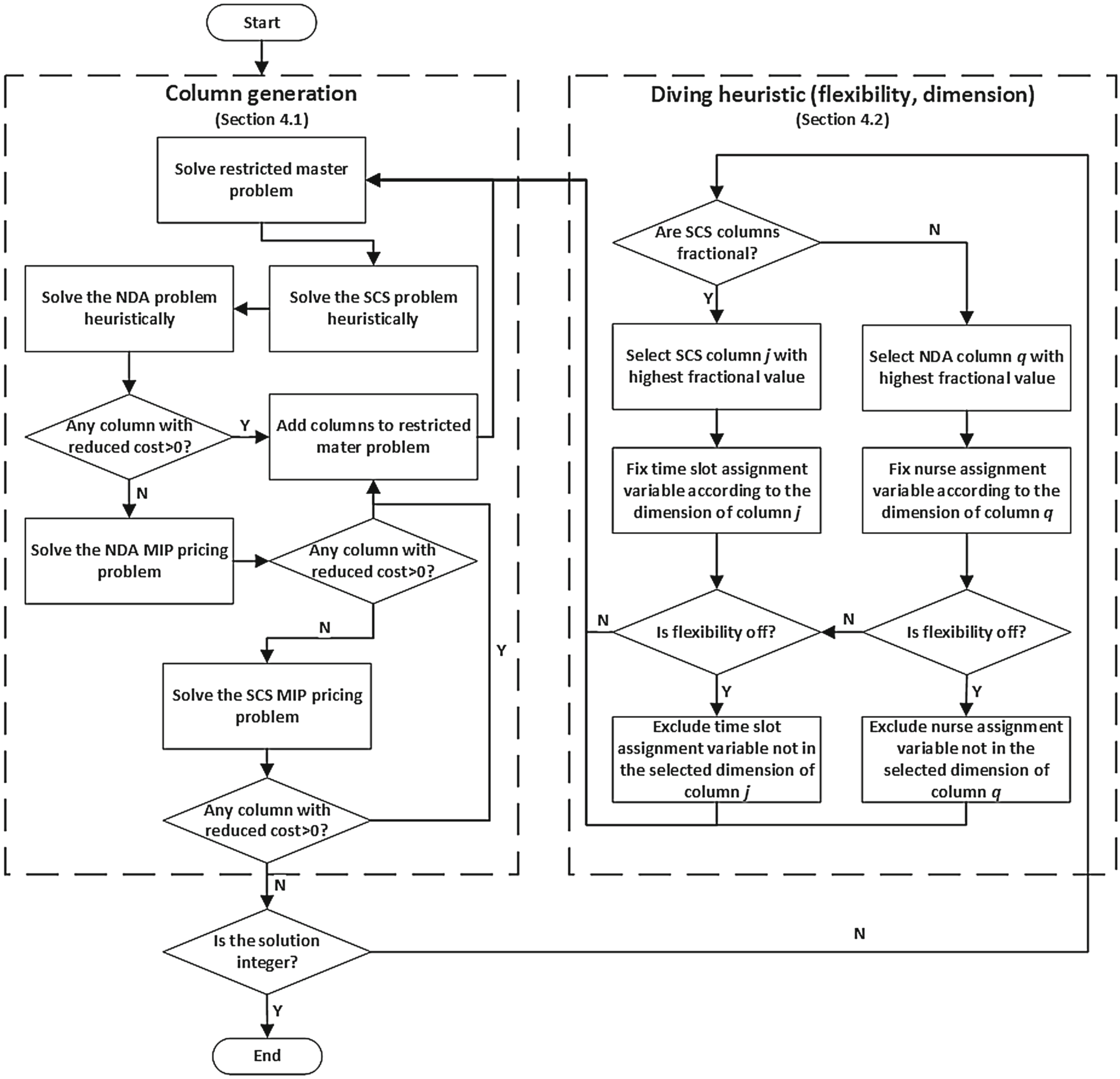

Start
Column generation
(Section 4.1)
Solve restricted master problem
Solve the NDA problem heuristically
Solve the SCS problem heuristically
Any column with reduced cost>0?
Y
N
Add columns to restricted mater problem
Solve the NDA MIP pricing problem
Any column with reduced cost>0?
N
Y
Solve the SCS MIP pricing problem
Any column with reduced cost>0?
N
Is the solution integer?
Y
End
N
Diving heuristic (flexibility, dimension)
(Section 4.2)
Are SCS columns fractional?
N
Y
Select SCS column j with highest fractional value
Select NDA column q with highest fractional value
Fix time slot assignment variable according to the dimension of column j
Fix nurse assignment variable according to the dimension of column q
N
Is flexibility off?
N
Is flexibility off?
Y
Y
Exclude time slot assignment variable not in the selected dimension of column j
Exclude nurse assignment variable not in the selected dimension of column q

Fig. 1 Flow chart of the column generation-based diving heuristic solution procedure (CGD)

problem. A column is added to the restricted master problem, when the column has a positive reduced cost as we consider a maximisation problem. The pricing step is conceived as a three-stage procedure. In the first stage, we try to find interesting SCS and NDA columns using a greedy heuristic. In the second stage, we solve the NDA pricing problem to optimality using mixed integer programming (MIP). In the third stage, we solve the SCS pricing problem to optimality using MIP. Whenever a column with a positive reduced cost has been found in a particular stage, we add this column to the restricted master problem and resolve the restricted master problem. If no suitable column has been found in any stage, the column generation procedure is stopped and we have found the optimal LP solution, which is an upper bound for the proposed problem. If the solution in this step is integer, we have found an optimal solution for the original integer problem. If not, we need additional steps to convert the fractional solution into a high-quality integer solution. In the *second step*, a greedy diving heuristic is applied to convert the fractional solution into a feasible integer solution, which is detailed in Sect. 4.2. The diving heuristic repetitively fixes one or several fractional assignments in a depth-first search without backtracking. In each node of the diving tree, the diving heuristic focuses first on the SCS column with the highest fractional value and fixes surgical case assignments to particular time slots according to the selected column. The amount of surgical case assignments that are fixed depends on the so-called dimension, which is a design parameter of the diving heuristic reflecting the involved schedule components (patient, room, day) for which assignments are fixed. Furthermore, this fixing may be related to either the original assignment variables considered in the pricing problem or to the column variables considered in the master problem. This is determined by the allowed flexibility in the search for a high-quality solution, which is a design parameter of the diving heuristic. When flexibility is not allowed, the fixing is performed identically to the selected column and no time slot assignments can additionally be included to the selected dimension later in the diving tree. Relevant assignment prohibition constraints are imposed avoiding the assignment of other surgical cases to the time slots related to the selected dimension. When flexibility is allowed, only the original assignment variables in the chosen dimension of the selected column are fixed and there is flexibility to include additional surgical cases to the selected dimension later in the tree. Whenever all SCS columns are integer, the diving heuristic considers the NDA columns with a fractional value and the same logic is applied fixing the assignments of surgical cases to specific nurses. After fixing a number of variables, column generation is resolved to find the LP optimal solution taking the diving constraints into account. When all SCS columns and NDA columns are integer, the diving heuristic returns an integer solution, which is the solution returned by this procedure and a lower bound of the optimal solution.

4.1 Column generation

In order to obtain the optimal LP solution, we relax the integrality conditions (Eq. 8) and apply a column generation algorithm iterating between the master problem and two pricing subproblems to identify interesting SCS and NDA columns that should be added to the master problem. The restricted LP master problem, defined by Eqs. (1)–(7), considers only a subset of the column variables by introducing the sets $\bar{J}_d$ $(\subset J_d)$ and $\bar{Q}_{ndv}$ $(\subset Q_{ndv})$. Next, the column generation procedure tries to identify new schedules to enter the basis based on the appropriate dual information resulting from the restricted master problem (Lübbecke and Desrosiers 2005). A column is added to $\bar{J}_d$ $(\bar{Q}_{ndv})$ when the column, obtained by solving the surgical (nurse duty assignment) pricing problem, has a positive reduced cost (cf. Eq. 1) (Sect. 4.1.1). When a column has been added, the master problem is invoked again. If the pricing step does not find a schedule to enter the basis, the column generation terminates and we have found the optimal LP solution. In order to improve the computational efficiency, we propose different speed-up techniques in Sect. 4.1.2. In this way, we first try to find a column in a heuristic manner. If these pricing problem heuristics are not able to come up with a column with a positive reduced cost, the pricing problems are solved to optimality via a commercial solver with a pricing cut included to speed up the performance. The pseudocode of the column generation procedure is presented in Algorithm 1.

Algorithm 1 Column generation

```
do
    Solve restricted master problem
    Solve SCS pricing problem heuristically ($\forall d$)
    Solve NDA pricing problem heuristically ($\forall n, d, v$)
    if No column found then
        Solve NDA MIP pricing problem ($\forall n, d, v$)
        if No column found then
            Solve SCS MIP pricing problem ($\forall d$)
    Add column with positive reduced cost to the restricted master problem
while SCS or NDA column with positive reduced cost exists
```

4.1.1 Pricing problem

The type of decomposition used to model the master problem in Sect. 3 determines the definition and formulation of the subproblem(s). In the proposed model formulation, two types of column variables are utilised, each defined by a specific subproblem and pricing problem in a column generation framework, i.e. the surgical case schedule pricing problem and the nurse duty assignment pricing problem. In order to formulate the subproblems, the following symbols are additionally defined, i.e.

Parameters

A_{strd} 1, if time slot t at day d in room r belongs to surgeon s according to the MSS; 0, otherwise

ROS_{ndv} 1, if nurse n is assigned to shift v, day d in the original nurse roster; 0, otherwise

Pr_{ndv} 1, if nurse n has preference to work shift v, day d; 0, otherwise

St_v^V Start time of shift v

Et_v^V Finish time of shift v

Du_v Duration of shift v

t_i Duration of surgery i

Decision variables

px_{itrd} 1, if surgery i starts at time slot t on day d in room r on day d; 0, otherwise

a_{id} 1, if surgical case i is selected on day d; 0, otherwise

av_{idv} 1, if surgical case i is performed in shift v, day d; 0, otherwise

e_{rd} 1, if operating room r is opened on day d; 0, otherwise

$P_{i\bar{i}d}$ 1, if surgical case i and $\bar{i}$ overlap on day d; 0, otherwise

h_{indv} 1, if surgical case i is selected for nurse n, day d, shift v; 0, otherwise

S_{id} Start time of surgery i on day d

C_{id} Finish time of surgery i on day d

$w_{i\bar{i}d}$ 1, if surgical case i starts before surgical case $\bar{i}$ on day d; 0, otherwise

$u_{i\bar{i}ndv}$ 1, if surgical cases i and $\bar{i}$ are both included on day d, shift v for nurse n; 0, otherwise

Note that the decision variables in the pricing problems define the parameters of the generated columns explained in Sect. 3 (e.g. a_{id} is a binary decision variable in the pricing problem that defines the column parameter $\bar{a}_{idj}$ for SCS column j).

The Surgical Case Schedule pricing problem

The SCS pricing problem is used to generate a column that represents a schedule of surgical cases in the different operating rooms. Instead of running the pricing problem for all days together, we decompose the pricing problem and generate schedules for each day separately to reduce the computational effort. The SCS pricing problem for day d is formulated as follows

$$\begin{aligned}\text{Max } RC_d^{\text{SCS}} = &+\sum_i Rev_i \times a_{id} - \sum_r c_r^{OR} \times e_{rd}\\ &- \sum_i a_{id} \times \pi_i^2 - \pi_d^3 - \sum_{iv} nu_i \times av_{idv} \times \pi_{idv}^6\\ &- \sum_{i\bar{i}} BM \times P_{i\bar{i}d} \times \pi_{i\bar{i}d}^7 \qquad (9)\end{aligned}$$

$$\sum_{tr} px_{itrd} \leq a_{id} \quad \forall i \qquad (10)$$

$$\sum_i px_{itrd} \leq 1 \quad \forall t, r \qquad (11)$$

$$\sum_{\bar{i} \neq i} \sum_{\bar{t}=t+1}^{\bar{t}=t+t_i} px_{\bar{i}\bar{t}rd} \leq BM(1 - px_{itrd}) \quad \forall i, t, r \qquad (12)$$

$$\sum_{it} px_{itrd} \leq BM \times e_{rd} \quad \forall r \qquad (13)$$

$$av_{idv} = \sum_{t=St_v^V, r}^{t=Et_v^V} px_{itrd} \quad \forall i, v \qquad (14)$$

$$S_{id} = \sum_{tr} t \times px_{itrd} \quad \forall i \qquad (15)$$

$$C_{id} = S_{id} + t_i \times \sum_{tr} px_{itrd} \quad \forall i \qquad (16)$$

$$S_{\bar{i}d} - S_{id} \leq BM \times (1 - w_{i\bar{i}d}) \quad \forall i, \bar{i} \qquad (17)$$

$$\begin{aligned}C_{\bar{i}d} - S_{id} \leq\ & BM \times P_{i\bar{i}d} + BM \times w_{i\bar{i}d}\\ &+ BM \times (2 - a_{id} - a_{\bar{i}d}) \quad \forall i, \bar{i} \qquad (18)\end{aligned}$$

$$t_i \times px_{itrd} \leq \sum_{\bar{t}=t}^{\bar{t}=t+t_i} A_{s\bar{t}rd} \quad \forall s, i \in \varphi_s, t \qquad (19)$$

$$\begin{aligned}&\sum_{ir} nu_i \times px_{itrd} \leq (ROS_{ndv} + Prf_{ndv})\\ &\quad \forall v, St_v^V \leq t \leq Et_v^V \qquad (20)\end{aligned}$$

$$\begin{aligned}&px_{itrd}, w_{i\bar{i}d}, P_{i\bar{i}d}, a_{id}, e_{rd} \in \{0, 1\} \quad \forall i, \bar{i}, t, r\\ &S_{id}, C_{id} \geq 0 \quad \forall i. \qquad (21)\end{aligned}$$

The objective function (9) maximises the reduced cost of a surgical case schedule. Constraint (10) stipulates that each surgery can be performed at most once on a specific day. Constraint (11) makes sure that at most one surgery can be assigned to a specific time slot and room. Constraint (12) states that a scheduled surgery cannot be interrupted by another surgery, i.e. only one surgery can be carried out in a specific room during a particular time slot. The total number of surgeries is a relevant upper bound in the right-hand side of the constraint, i.e. $BM = |I|$. Constraint (13) assigns surgeries to a room only if the operating room is open for the day under consideration. The large number BM can be operationalised by the total number of surgeries $|I|$.

Constraints (14)–(18) are introduced in the model formulation to include the appropriate dual information from the master problem and to generate suitable columns. Constraint (14) finds the shift duty during which the surgery is performed based on the assigned start time slot of the surgical case. Equations (15)–(18) determine if two surgical cases overlap. Constraints (15) and (16) calculate the start and completion time for each surgery. Constraint (17) calculates $w_{i\bar{i}d}$ for surgical cases i and $\bar{i}$. The value of the variable $P_{i\bar{i}d}$, which indicates if an overlap exists between surgical cases i and $\bar{i}$, is determined in constraint (18). If surgical case i starts later than case $\bar{i}$ and the completion time of case $\bar{i}$ is later than the start time of case i, there is an overlap between the two surgeries, i.e. $P_{i\bar{i}d} = 1$. In case the surgical case i starts before $\bar{i}$ (i.e. $w_{i\bar{i}} = 1$) or both surgeries are not selected, the constraint is redundant. The large number BM can be operationalised by the total number of time slots $|T|$ in Constraints (17) and (18).

Constraint (19) states that a surgery can be conducted only when the associated surgeon is available. Constraint (20) ensures that the scheduling of the surgical cases is in line with the maximum available number of nurses that can be scheduled during shift v according to the original roster and the nurse preferences. Constraint (21) states the integrality conditions and the domain of the variables.

The nurse duty assignment pricing problem

The NDA pricing problem is used to generate a column that composes a particular shift duty by assigning surgical cases to a specific nurse. Instead of running the pricing problem once, we decompose this pricing problem and generate schedules for each day, nurse and shift separately to reduce the computational effort. The NDA pricing problem for nurse n, day d and shift v is formulated as follows

$$\text{Max } RC_{ndv}^{\text{NDA}} = -c_v^N - \pi_n^4 - \pi_{nd}^5 + \sum_i h_{indv} \times \pi_{idv}^6 - \sum_{i\bar{i}} u_{i\bar{i}ndv} \times \pi_{i\bar{i}d}^7 \quad (22)$$

$$u_{i\bar{i}ndv} \le h_{indv} \qquad \forall i, \bar{i} \quad (23)$$

$$u_{i\bar{i}ndv} \le h_{\bar{i}ndv} \qquad \forall i, \bar{i} \quad (24)$$

$$u_{i\bar{i}ndv} + 1 \ge h_{indv} + h_{\bar{i}ndv} \qquad \forall i, \bar{i} \quad (25)$$

$$\sum_i t_i \times h_{indv} \le Du_v \quad (26)$$

$$\sum_i h_{indv} \le BM(ROS_{ndv} + Prf_{ndv}) \quad (27)$$

$$h_{indv}, u_{i\bar{i}ndv} \in \{0, 1\} \qquad \forall i, \bar{i}. \quad (28)$$

The objective function (22) maximises the reduced cost of the NDA column by assigning the best possible combination of surgical cases to the considered shift duty and nurse. In order to include the appropriate dual information from the master problem and generate suitable columns, we calculate if surgical cases i and $\bar{i}$ are both included in the nurse duty assignment, i.e. $u_{i\bar{i}ndv} = 1$. The AND relationship between the assignment variables h_{indv} and $h_{\bar{i}ndv}$ is modelled by equations (23)–(25). Constraint (26) implies that the duration of the assigned surgical cases cannot be more than the duration of the involved shift. Constraint (27) makes sure only nurse duty assignments are generated for a shift duty v on day d that is conform to the original roster or the nurse preferences of nurse n, i.e. $ROS_{ndv} + Prf_{ndv} = 1$. It is important to point out that the nurse preferences Prf_{ndv} determine the degree to which the original nurse roster can be changed. Equation (28) embodies the binary conditions of the variables.

4.1.2 CG speed-up techniques

Cut for pricing problems

When a schedule is created in the pricing step, the main idea is to incorporate those surgical cases that are able to improve the reduced cost of the column. For each individual surgery i on day d and shift v, the maximum reduced cost RCi_{idv}^{SCS} can be calculated based on Eq. (29), which includes the revenue of the surgery, the relevant dual prices $(\pi_i^2, \pi_{i,d}^9)$ to incorporate surgery i in the schedule and the dual prices $(\pi_i^7,\ \pi_i^8)$ associated with the related shift v.

$$RCi_{idv}^{\text{SCS}} = Rev_i - \pi_i^2 - \pi_d^3 - nu_i \times \pi_{idv}^6 \quad (29)$$

Lemma 1 *For any $i,\ d,\ v:\ RCi_{idv}^{\text{SCS}} \le 0$, scheduling surgical case i on day d during the shift v will lead to a decrease in the reduced cost. Therefore, without losing optimality, Eq. (30) can be added to the proposed SCS pricing problem.*

$$\sum_r \sum_{t=St_v^V}^{t=Et_v^V} px_{itrd} = 0 \qquad \forall\, i,\ v,\ d:\ RCi_{idv}^{\text{SCS}} \le 0. \quad (30)$$

Likewise, the reduced cost RCi_{idv}^{NDA} resulting from assigning surgery i to shift v on day d can be calculated (Eq. 31), i.e.

$$RCi_{idv}^{NDA} = \pi_{idv}^6. \quad (31)$$

Therefore, without losing optimality, Eq. (32) can be added to the proposed NDA pricing problem.

$$h_{indv} = 0 \qquad \forall\, i,\ v,\ d:\ RCi_{idv}^{i,NDA} \le 0. \quad (32)$$

Heuristic procedure for generating columns

Whenever the restricted master problem has been solved, the proposed CG starts the generation of SCS columns in a heuristic manner. The main idea is to incorporate those surgical cases with a negative individual reduced cost calculated in Eq. (29). Using this insight, the heuristic SCS columns are generated for each day separately. For each day, the algorithm schedules surgical cases based on descending order of RCi_{idv}^{SCS} until all surgical cases with positive RCi_{idv}^{SCS} are scheduled or the capacity of operating rooms is filled. During the scheduling of these surgical cases, the master surgery schedule is considered. Afterwards, the reduced cost for the created SCS column is calculated. The columns with positive reduced cost are added to the restricted master problem. Likewise, we used Eq. (31) to generate the heuristic NDA columns for each nurse, day and shift.

4.2 Diving heuristic

In order to transform the optimal LP solution resulting from the column generation into an integer solution, we propose a diving heuristic procedure. Diving heuristics have been successfully implemented in the literature for different applications. Joncour et al. (2010) and Sadykov et al. (2019) give an overview of different diving techniques to find a heuristic solution and possibly its neighbouring solutions. The diving heuristic repetitively fixes particular assignments and uses column generation in each iteration. The diving heuristic applies a depth-first search without backtracking until an integer solution is found by fixing at each node of the tree one or several fractional variables to a positive integer value. For decomposed problem formulations solved via branch-and-price such a variable may be either an original assignment variable, i.e. a pricing problem variable that is defined to formulate the pricing problem, or a column variable, i.e. a master problem variable. Once a variable is fixed, the LP is re-optimised taking these diving constraints into account. In this way, the diving heuristic performs a heuristic search in an LP-based branch-and-bound tree. Algorithm 2 presents the proposed diving heuristic procedure, which is explained below.

Algorithm 2 Diving Heuristic (Flexibility, Dimension)

```
Run Column generation
while The column generation solution is fractional do
    if The SCS columns are fractional then
        Select SCS column $j^*$ with the highest fractional value
        if Flexibility is allowed then
            Fix $px_{itrd} = 1 \ \forall i : \bar{a}_{idj} = 1$ in the selected Dimension of column $j^*$
        else
            Fix $px_{itrd} = 1 \ \forall i : \bar{a}_{idj} = 1$ in the selected Dimension of column $j^*$
            Fix $px_{itrd} = 0 \ \forall i : \bar{a}_{idj} = 0$ in the selected Dimension of column $j^*$
    else
        Select NDA column $q^*$ with the highest fractional value
        if Flexibility is allowed then
            Fix $h_{indv} = 1 \ \forall i : \bar{h}_{indvq} = 1$ in the selected Dimension of column $q^*$
        else
            Fix $h_{indv} = 1 \ \forall i : \bar{h}_{indvq} = 1$ in the selected Dimension of column $q^*$
            Fix $h_{indv} = 0 \ \forall i : \bar{h}_{indvq} = 0$ in the selected Dimension of column $q^*$
    Run Column generation
End while
```

In order to determine the variables that are fixed at each node of the tree, the algorithm focuses first on the surgical case planning and scheduling problem by selecting the SCS column j^* ($j^* \in J_d, \forall d$) over all days with a fractional value that is closest to 1. When the surgical case schedule is integer, the NDA column q^* ($q^* \in Q_{ndv}, \forall n, d, v$) with a fractional value closest to 1 is selected. This branching priority is established as in most cases an integer surgical case schedule leads to a set of integer NDA columns. In the proposed diving heuristic, these selected columns are (partially) fixed via assigning the included surgical cases to particular time slots for SCS columns and to specific nurses for NDA columns. In order to fix specific assignment variables according to the selected column, the proposed diving heuristic thrives on two concepts, i.e.

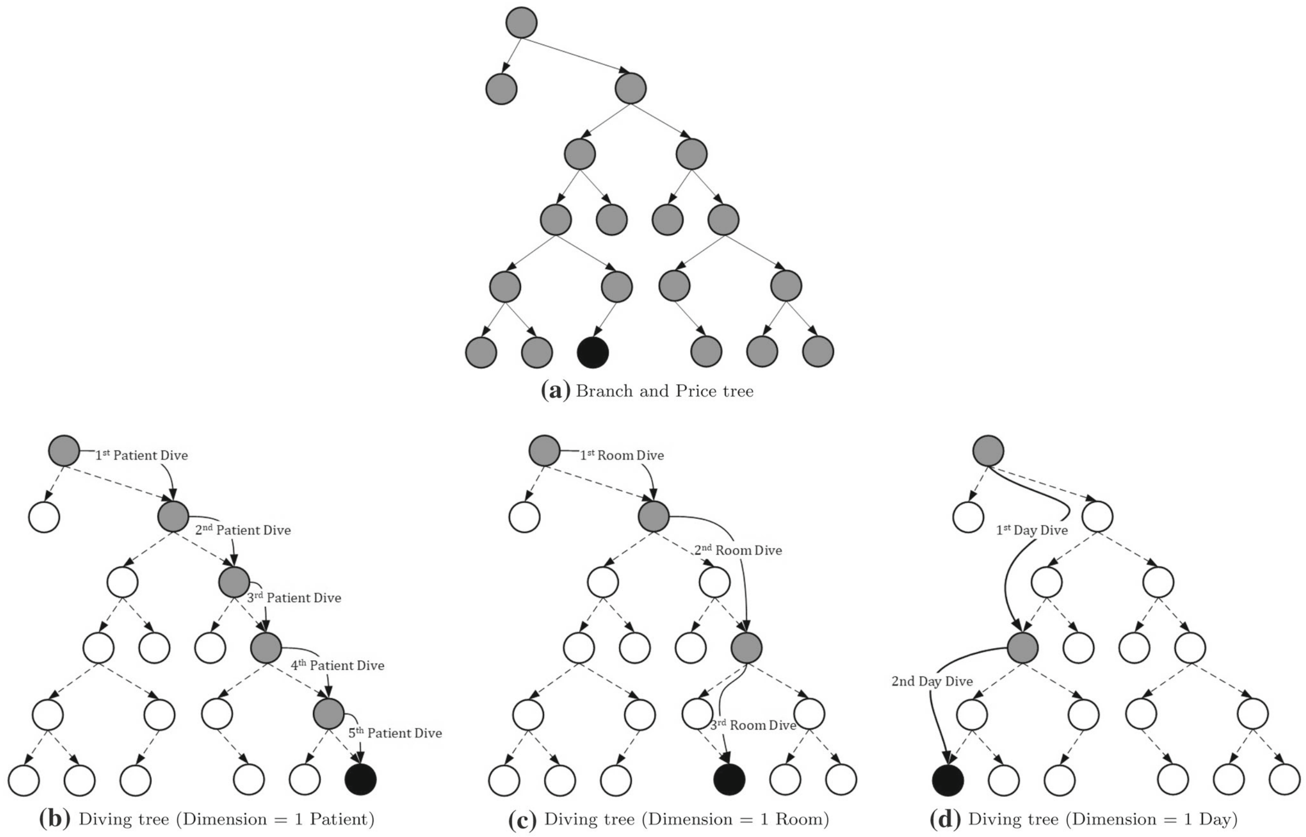
Dimension

The selected dimension refers to the components of the surgical case schedule or nurse duty assignment schedule for which variables are fixed. The dimension is defined by the dimension characteristic and the dimension size.

- The *dimension characteristic* refers to the feature of the selected SCS column j^* or NDA column q^* that is considered for variable fixing and may involve components at the level of the patient, room and day for SCS columns and components at the level of the shift for NDA columns.
- The *dimension size* is defined as the number of components of the relevant dimension characteristic for which variables are fixed. The components are prioritised and selected based upon the number of surgical assignments included, i.e. the higher the number of surgical assignments, the higher the priority. E.g. if the dimension embodies two days, two days are selected out of the planning horizon based upon the number of surgical assignments.

Only the surgical cases of the selected components are fixed according to SCS column j^* or NDA column q^*. Hence, when the selected dimension of an SCS column is different than a single day, we do not fix all surgical cases included in column j^*. In contrast to traditional diving heuristics, this feature allows us to only partially fix the assignments of the selected column. Using this concept, we can control the number of surgical case assignments fixed at each node of the tree. Downgrading the dimension increases the number of nodes visited and the depth of the tree.

Figure 2 illustrates the diving tree using a patient, an operating room and a day as the relevant dimension and a branch-and-price tree. In these figures, the grey nodes represent the visited nodes, the black nodes refer to the best incumbent solution found and the white nodes are indicated in the diving trees to make the comparison with the branch-and-price tree more apparent as these nodes are not visited in the diving trees. The branch-and-price tree (Fig. 2a) consists of a large number of grey nodes visited, and the branching is typically done based on the patient level. Figure 2b represents a diving tree using one patient as relevant dimension. Note that the diving tree visits one node on each level of the branch-and-price tree. Figure 2c, d represents the diving trees with one operating room or one day respectively as dimension. These dimensions comprise multiple patient assignments such that not every layer in the branch-and price tree is visited. The diving tree in Fig. 2c thriving on the operating room as dimension has a depth of three assignments, indicated by the number of branches with solid lines, to find an integer incumbent solution, whereas the diving tree in Fig. 2d using the day as dimension characteristic needs only two assignments to find an incumbent solution.



(a) Branch and Price tree

(b) Diving tree (Dimension = 1 Patient) **(c)** Diving tree (Dimension = 1 Room) **(d)** Diving tree (Dimension = 1 Day)

Fig. 2 Illustration of the visited nodes in (a) a branch-and-price tree and diving trees with dimension characteristic (b) Patient (c) Room (d) Day

Flexibility

The variables included in the selected dimension may refer to either the involved pricing problem variables or the master problem variable indicating whether flexibility is respectively allowed or not in the further search for a heuristic solution.

- **Flexibility = OFF**
  In case variables are fixed according to the involved *master variable*, all surgical cases related to the selected dimension are assigned to a particular time slot listed by SCS column j^*, i.e. $px_{itrd} = 1$. The assignment of other surgeries is not allowed for the considered dimension of column j^*, and relevant assignment prohibition constraints, i.e. $px_{itrd} = 0$, are imposed. In this way, the surgical case schedule is identical to the schedule reflected by the selected column j^* and additional flexibility is not granted. When the dimension embodies an entire column, i.e. a single day and all operating rooms, the variable fixing according to the involved master variables resembles the classic diving heuristic that fixes an entire column variable at each node of the tree.

- **Flexibility = ON**
  In case reference is made to the involved *pricing variables* only, all surgery assignments to a particular time slot listed by SCS column j^* are fixed, i.e. $px_{itrd} = 1$. If possible, other surgical assignments can be added to the selected dimension in the subsequent search for an integer solution. E.g. if the dimension involves an operating room, all surgeries performed in this operating room according to column j^* are fixed and other surgical cases may be included in the surgery timetable for the operating room in subsequent nodes of the diving tree. In this way, flexibility is installed in the composition of an SCS column.

When an NDA column is selected, a similar reasoning can be applied. All surgical cases listed by column q^* related to the selected dimension are assigned to a particular nurse, i.e. $h_{indv} = 1$. In case the variables are fixed according to the involved master variable, no additional assignments are allowed and assignment prohibition constraints are imposed, i.e. $h_{indv} = 0$ for all other assignments of surgical cases to nurses.

Table 2 Parameter settings characterising the problem size

Parameter	Description	Settings
Avg_I	Average total number of surgical cases in the planning horizon	10–15–20–30
$\|R\|$	The number of operating rooms	2–4
$\|V\|$	The number of available shifts on a particular day	1–3
$\|T\|$	The number of time slots on a particular day	32
$\|D\|$	The number of days in the planning horizon	2

5 Computational experiments

In this section, the computational performance of the proposed methodology is verified. In Sect. 5.1, we discuss the test design by describing the test instances and parameter test settings. Section 5.2 demonstrates the performance of every component of the algorithm and benchmarks the proposed procedure with an MIP model solved via the commercial optimisation package. All mathematical models are solved using CPLEX 12.7 with default optimisation parameters. The proposed algorithm was coded in C#, and all tests were carried out on an Windows PC with Corei3 - 2.50 GHz CPU and 4 GB RAM.

5.1 Test design

In order to study the performance of the proposed solution approach, we construct a set of problem instances by varying the input parameters in a systematically varied and controlled way. The considered parameters and their settings are based upon the observation of real-life problem environments and preliminary experiments. Table 2 presents the parameters characterising the problem size. For computational reasons, we limit the number of days in the planning horizon to 2. Note that the (limited) number of operating rooms is related to the considered cluster of surgeons or medical disciplines for which the resources (OR, nurse) can be exchanged.

Based on the proposed parameter settings in Table 2, we compose a full factorial design of experiments leading to 16 different classes of instances. For each class, we generated 20 instances resulting in 320 instances in total. The other parameters are parameterised for each instance as follows, i.e.

Surgical cases

The *surgery duration* t_i is determined based upon a discrete uniform distribution from the interval $[1, 4]$. The minimum (maximum) surgery duration of 1 (4) time slot(s) is selected based on the relation with the maximum (minimum) number of surgical cases carried out in the hospital under investigation. Note that this generation is done for different types of surgeries together and the real-life surgery durations could not be represented by a single continuous (Log-Normal) distribution, which was proposed by Day et al. (2012).

Surgeons

The *number of surgeons* $|S|$ is uniformly selected from the interval $[|R|, |R| \times |D| \times BFctr]$. The lower bound represents the case where the same number of surgeons is present on each day, equal to the number of operating rooms. The upper bound is determined based upon the number of surgeons that can share an operating room, i.e. the number of blocks on each day represented by *BFctr*. Based upon real-life experience, we set $BFctr$ equal to 2 resulting in a morning and afternoon block (Blake and Donald 2002). The number of surgical cases on a surgeon list are determined randomly based on a Poisson distribution with $\lambda = \frac{Avg_I}{|S|}$. The sum of all patients on the surgeon lists (i.e. $\sum_s |\varphi_s|$) results in the total number of surgical cases $|I|$ considered.

The *master surgery schedule*, which characterises the input parameter A_{strd}, is constructed by assigning the available time slots equally to the surgeons, i.e. $\frac{|R|\times|D|\times|T|\times BFctr}{|S|}$ time slots per surgeon. The assignment procedure starts by randomly selecting a surgeon, day and time slot related to the start of a block. Starting from this time slot, the time slots for the surgeon are chronologically assigned in a consecutive manner. This procedure is continued until all time slots of all surgeons have been assigned.

Nursing staff

The *number of nurses* $|N|$ is uniformly selected from the interval $[|R| \times |V|, |R| \times |V| \times |D|]$. The minimum (maximum) required number of nurses is $|R|\times|V|$ ($|R|\times|V|\times|D|$), i.e. each day the same (a different) set of nurses is operating. The originally announced nurse roster (ROS_{ndv}) and the nurse preferences (Prf_{ndv}) are complementary defined such that the nurses have the complete flexibility to be assigned to any shift and day and focus is given to the minimisation of the number of nurse shift duties. Note that this is not the case for the experiments conducted on real-life data in Section 6.

The *work organisation* of nurses is done according to non-overlapping shifts. The *duration* of each shift Du_v is fixed and equals $\frac{|T|}{|V|}$. $|V|$ shifts of equal length are created starting from the opening time of the OR department until closing time.

Objective function

The objective function structure is determined in dialogue with the visited Sina Hospital (Tehran, Iran) (cf. Sect. 6). The relation between the different objective components is as follows, i.e. we set the $Rev_i = 1000$, $c_r^{OR} = 100$ for any operating rooms and $c_v^N = v \times 10$, which implies later shifts are more costly than earlier shifts.

5.2 Algorithm performance

In this section, we give insight into the computational performance of the different components of the proposed solution methodology. In Sect. 5.2.1, we analyse and benchmark the general performance of the CGD procedure with an MIP model solved via the commercial solver CPLEX with a time limit of 7200 s. The MIP model is included in "Appendix A". In Sect. 5.2.2, we study the performance of the column generation and the impact of the proposed speed-up techniques. Section 5.2.3 discusses different variants of the diving heuristic in order to get insight into the required settings for diving related to the dimension and the flexibility. In the remainder, we report the results on all our tests using the following symbols, i.e.

Obj_x the solution value (cf. Eq. 1) obtained by procedure x

$\%Dev_x$ the lower bound deviation percentage between the integer solution value obtained by procedure x and the MIP model (= $\frac{(Obj_{MIP} - Obj_x)}{Obj_{MIP}}$)

$\%Dev_x^{LP}$ the upper bound deviation percentage between the LP solution value obtained by procedure x and the MIP model (= $\frac{(Obj_{MIP}^{LP} - Obj_x^{LP})}{Obj_{MIP}^{LP}}$)

$\%Gap_x$ the relative optimality gap between the integer solution value obtained by procedure x and the LP solution value obtained via column generation (= $\frac{(Obj_{CG} - Obj_x)}{Obj_{CG}}$)

$\%OutP_x$ the percentage of instances that procedure x performs better solution than the MIP procedure
$\%Opt_x$ the percentage of instances solved to optimality ($\%Gap_x = 0$) by procedure x
CPU_x the required computational time in seconds for procedure x
CPU_y^{Pr} the required computational time in seconds for solving pricing problem of type *y*
$\#Br_x$ number of branches of type that are added during the procedure x
$\#clmn_y$ number of columns of type *y* added via pricing problem
$\#clmn_y^{Heur}$ number of columns of type *y* added by using heuristic procedure (cf. Sect. 4.1.2)

with $x = MIP$ the benchmark MIP solution procedure using the commercial solver CPLEX solving the model in "Appendix A" with a time limit (7,200 s)

$x = CG$ the proposed column generation

$x = DH$ the diving heuristic

$x = CGD$ the proposed Column Generation based Diving heuristic

with $y = SCS$ related to the SCS columns

$y = NDA$ related to the NDA columns

$y = -$ related to both SCS and NDA types of columns

5.2.1 General performance of CGD

We first analyse the general performance of the proposed solution methodology and its building blocks. The results of the computational experiments are summarised in Table 3 based on the parameters characterising the problem dimensions. We display the results for the column generation (CG), the best performing diving heuristic (DH) with a dimension equal to one day and flexibility allowed, the entire procedure (CGD) and the MIP model.

Table 3 shows that when the average number of patients (Avg_I) is increased, the required CPU of the MIP solution procedure (CPU_{MIP}) and the proposed procedure (CPU_{CGD}) rises as a result of the increase in complexity of the surgical case planning and scheduling problem. The same effect is observed when the number of shifts ($|V|$) rises. The latter can be explained by the increased complexity of the nurse re-rostering problem as a result of the larger problem size and the objective function structure, which primarily tries to assign nurses to the earlier shifts due to the lower cost. Only when the number of operating rooms ($|R|$) increases, the required CPU time is lower as a result of the inadequate OR capacity. A lower number of rooms increases the resource tightness and the complexity of selecting the appropriate set of surgical cases.

When analysing the performance of the *column generation* and the postulated decomposed problem formulation, we observe the same effects of the problem size parameters on the CPU time. A higher value for the factors Avg_I and $|V|$ ($|R|$) increases (decreases) the required CPU_{CG}. Note that when the problem complexity is limited, the col-

Table 3 General performance of the proposed CGD procedure

	Ave_I				$\|R\|$		$\|V\|$		*Overall*
	10	15	20	30	2	4	1	3	
CG									
CPU_{CG}	9	19	32	490	247	28	98	177	137
$\#clmn_{-}$	986	2498	3084	8607	3807	3781	2448	5140	3794
$\%Opt_{CG}$	15%	20%	0%	0%	5%	12%	12%	5%	9%
$\%Dev^{LP}_{CGD}$	4.37%	3.45%	3.70%	4.63%	4.68%	3.40%	4.80%	3.28%	4.04%
CGD									
CPU_{CGD}	22	47	105	813	397	97	196	297	247
$\%Dev_{CGD}$	0.02%	0.02%	0.02%	−0.84%	−0.41%	0.02%	−0.09%	−0.30%	−0.19%
$\%Gap_{CGD}$	0.02%	0.02%	0.27%	0.41%	0.29%	0.08%	0.20%	0.16%	0.18%
$\%Opt_{CGD}$	35%	45%	15%	20%	30%	28%	50%	7%	29%
MIP									
CPU_{MIP}	84	1904	1956	4752	2458	1891	1294	3054	2174
$\%OutP_{CGD}$	45%	50%	65%	90%	55%	70%	97%	27%	62%

umn generation finds already the optimal integer solution for some of the instances ($\%Opt_{CG}$). The decomposed formulation strengthens the LP formulation by more than 4% on average ($\%Dev^{LP}_{CGD}$). The computational effort to build an integer solution using the *diving heuristic* starting from the fractional LP solution can be deducted by comparing CPU_{CGD} with CPU_{CG}. This effort averages 109 s over all instances and increases with the problem complexity. The CGD heuristic solves 29% of the instances to optimality in a limited time span. In general, the proposed algorithm delivers solutions close to the optimal solution as the optimality gap is on average only 0.18% ($\%Gap_{CGD}$). The optimality gap is higher for especially those larger instances with on average 20 or 30 surgical cases. The proposed procedure outperforms the MIP procedure in terms of solution quality and CPU time. The average time to solve the instances to optimality is 247 s (CPU_{CGD}). Note that this effort is significantly smaller compared to the MIP procedure, which amounts on average 2174 s (CPU_{MIP}). In terms of solution quality, the deviation $\%Dev_{CGD}$ equals -0.19%, denoting that on average the proposed procedure returns a better schedule quality than the MIP procedure, although the MIP procedure uses a time limit of 7200 s. As a result, the percentage of instances the proposed procedure outperforms the MIP procedure ($\%OutP_{CGD}$) amounts 62% and is significantly higher for larger instances with on average 20 or 30 surgical cases.

5.2.2 Performance of the column generation step

The impact of the speed-up techniques in the column generation step is determined via the comparison of different variants of the proposed algorithm. Each variant excludes one specific speed-up technique leaving the rest of the algorithm unchanged. In this way, we leave out the generation of the heuristic SCS columns (*w/o Heur SCS*); the generation of the heuristic NDA columns (*w/o Heur NDA*); the SCS pricing problem cut (*w/o Cut SCS*); and the NDA pricing problem cut (*w/o Cut NDA*). We also considered a version that includes all speed-up techniques (*All*). Table 4 displays the results of these variants. Note that the results are averaged over all instance groups.

Overall, Table 4 reveals that the removal of a speed-up technique leads to a worse performance in terms of the computational effort (CPU_{CGD}) compared to the proposed algorithm (*All*). The computational effort to conduct the column generation in every node of the diving tree and in particular in the pricing step (CPU^{Pr}_{SCS} and CPU^{Pr}_{NDA}) has increased when a speed-up technique is removed. This is in particular due to the generation of a higher number of columns via the MIP pricing step in the column generation process ($\#clmn_{SCS}$ and/or $\#clmn_{\text{NDA}}$). If we leave out the generation of heuristic SCS columns (NDA columns), note that the number of columns constructed during the pricing step rises from 54 to 185 (from 1558 to 2225). Leaving out the SCS pricing cut (‘*w/o Cut SCS*’) or the NDA pricing cut (*w/o Cut NDA*) increases the CPU time of the pricing step considerably due to the search for a good column in a larger solution space.

The exclusion of a speed-up technique has no impact on the solution quality ($\%Dev_{CGD}$) because of the strong performance of the diving heuristic that is able to come up with final solutions close to optimality. The removal of the fast generation of heuristic columns (*w/o Heur SCS* and *w/o Heur NDA*) restricts the number of eligible columns and leads to a set of columns with similar solution quality generated via the pricing problem. As a result, the diving heuristic is limited in its selection in the nodes of the diving tree to find a feasible

Table 4 Impact of speed-up techniques

	w/o *Heur SCS*	*w/o* *Heur NDA*	*w/o* *Cut SCS*	*w/o* *Cut NDA*	*All*
CG					
$\#clmn_{\text{NDA}}^{Heur}$	304	0	1955	2015	2045
$\#clmn_{\text{NDA}}$	2076	2225	1553	1541	1558
$\#clmn_{SCS}^{Heur}$	0	111	135	136	136
$\#clmn_{SCS}$	185	60	56	53	54
CPU_{NDA}^{Pr}	29	53	18	64	16
CPU_{SCS}^{Pr}	184	146	172	143	150
DH					
$\#Br_{DH}$	3.44	3.46	3.46	3.16	3.23
CPU_{DH}	73	71	112	82	84
CGD					
CPU_{CGD}	251	263	249	288	247
$\%Dev_{CGD}$	−0.19%	−0.19%	−0.19%	−0.19%	−0.19%
$\%Gap_{CGD}$	0.18%	0.18%	0.18%	0.18%	0.18%

solution and requires more branching (Br_{DH}) leading to a larger diving tree with a higher number of visited nodes.

5.2.3 Impact of the parameter settings for the diving heuristic

In order to obtain insight into the performance of the diving heuristic, we vary the settings for the parameters dimension and flexibility, described in Sect. 4.2. Table 5 displays the results for the most relevant settings, i.e.

– **Dimension = All days, Flexibility = OFF**
  The diving heuristic selects the most fractional column for every day and fixes these entire columns to a positive integer value. These parameter settings result in an integer surgical case schedule in the root node of the diving tree. Dimension = One day, Flexibility = OFF
  At each node of the diving tree, the diving heuristic selects the most fractional column in the planning horizon and assigns this entire column to a positive integer value, fixing the surgical case schedule for one particular day.
– **Dimension = One day, Flexibility = ON**
  At each node of the diving tree, the diving heuristic selects the most fractional column in the planning horizon and assigns the surgical cases contained in this column to their indicated time slots. Note that the surgical case schedule of the involved day can be modified later in the diving tree by including additional surgical cases.
– **Dimension = One room, Flexibility = ON**
  At each node of the diving tree, the diving heuristic selects the most fractional column in the planning horizon and assigns the surgical cases contained in this column to their indicated time slots only for the operating room with the highest number of surgical cases. Note that the surgical case schedule of the involved room and day can be modified later in the diving tree by including additional surgical cases.
– **Dimension = One patient, Flexibility = OFF**
  At each node of the diving tree, the diving heuristic selects the surgical case with the highest number of different start times according to the columns in the LP solution. This surgical case is assigned to the time slot indicated by the column with the largest fractional value, closest to 1. This assignment is fixed and cannot be modified in later nodes of the diving tree.

These versions of the diving heuristic are compared with a branch-and-price procedure that branches upon the assignment of individual patients (i.e. the dimension equals one patient). In contrast to the diving heuristic, the branch-and-price is an optimal procedure for which backtracking is possible. Hence, as assignments can be changed later in the branch-and-price tree, flexibility is allowed.

Overall, Table 5 reveals that a larger dimension decreases the computational effort (CPU_{CGD}) as a larger dimension decreases the number of created branches in the diving tree and thus the number of visited nodes ($\#Br_{CGD}$). A larger dimension, however, leads a lower solution quality ($\%Dev_{CGD}$). The solution quality is best when the dimension is set at the level of an individual patient and is worst when the dimension embodies all days. Note that there is no difference when the dimension is set to one day or one room.

If surgical case schedules can be modified, i.e. flexibility is allowed, the required computational effort increases but the performance ($\%Dev_{CGD}$) of the heuristic improves. Adding

Table 5 Impact of different dimensions and flexibility on the diving heuristic

	Diving heuristic					Branch and Price
Specification						
Dimension	All days	One day	One day	One room	One patient	One patient
Flexibility	OFF	OFF	ON	ON	OFF	ON
DH						
$\#clmn_{-}$	1628	2015	3794	6391	4050	26023
$\#Br_{DH}$	0.0	3.3	4.2	5.3	12.9	95.5
$\%Dev_{DH}$	0.71%	0.19%	−0.19%	−0.19%	−0.23%	−0.24%
$\%Gap_{DH}$	1.08%	0.56%	0.18%	0.18%	0.14%	0.00%
CGD						
CPU_{CGD}	151	175	247	288	707	4024
$\%Dev_{CGD}$	0.71%	0.19%	−0.19%	−0.19%	−0.23%	−0.24%
$\%Gap_{CGD}$	1.08%	0.56%	0.18%	0.18%	0.14%	0.00%

flexibility increases the number of branches and the number of columns generated.

The branch-and-price attains the optimal solution for every instance ($\%Gap_{CGD} = 0.00\%$) and leads naturally to the best solution quality. However, due to the large number of generated columns and created branches, the CPU time is significantly higher (4024 s). As a result, based upon these considerations, we propose a diving heuristic characterised by a dimension of a single day whereas flexibility is allowed.

6 Validation on real-life case study

The performance of the proposed algorithm has been tested and validated on a real-life case study involving the Sina Hospital (Tehran, Iran). The Sina Hospital is one of the largest and most prominent hospitals in Iran accounting for 338 beds and more than 700 staff members. More than 7000 surgical cases are operated each year in the operating room department. In this section, we discuss the input data provided by the hospital (Sect. 6.1) and the results of our experiments (Sect. 6.2). The goal of the computational experiment on the real-life case study is threefold. First, we want to verify if the proposed algorithm obtains an acceptable performance when real-life data is considered. Second, we evaluate the potential of the proposed approach by making the comparison with the current scheduling practices at the hospital. Third, we demonstrate the value of integrating the surgical case planning and scheduling problem with the nurse re-rostering and surgical case assignment by assessing the solution quality of the integrated approach.

6.1 Input data real-life case study

The hospital has several operation wards and provided us with information about the neurosurgery operating ward in particular related to its organisation, surgeries and staff, i.e.

Organisation

This ward contains four operating rooms and one additional operating room for emergency surgeries. Each time slot represents 30 minutes, dividing each day in 48 time slots.

Surgical cases

The provided surgery data relate to the surgery duration and the nurse requirements. The surgery duration, which is represented by a number of time slots, is determined by rounding the actual duration to the closest integer number of time slots. The nurse requirements, obtained from real-life data, vary case per case in the range $[1,\ 3]$.

Surgeons

The master surgery schedule embodies a timetable for 21 different surgeons.

Nursing staff

There are in total 40 surgical nurses in this ward who perform operations in three different shifts. The nurse roster is taken from the hospital. The nurse preferences were discussed with the head nurse responsible for preparing the nurse roster.

The obtained data are related to 10 weeks from January to April in 2016. In accordance with the hospital manager, we set the relevant planning horizon for the problem under study to five days, i.e. the problem is solved on a weekly basis for the next upcoming week. The objective function structure is already discussed in Sect. 5.1.

6.2 Validation of the algorithm on real-life data

Table 6 shows the results for the proposed methodology applied to the real-life case study. We present the result for each week and the average results for all weeks. We display the number of surgeries ($|I|$), the computational effort (CPU_{CGD}) and the solution quality via the optimality gap ($\%Gap_{CGD}$). In addition, we make the comparison between the proposed methodology (CGD) and the cur-

Table 6 The performance of CGD on the real-life data

i	$\|P\|$	CPU_{CGD}	$\%Gap_{CGD}$	$\sum_{nd,v=1} nx_{ndv}$		$\sum_{nd,v=2} nx_{ndv}$		$\sum_{nd,v=3} nx_{ndv}$		$\sum_{ndv} nx_{ndv}$		$\sum_{rd} e_{rd}$	
				CGD	*Cur*	*CGD*	*Cur*	*CGD*	*Cur*	*CGD*	*Cur*	*CGD*	*Cur*
1	60	4157	0.48	24	39	17	26	9	17	50	82	17	24
2	59	4020	2.38	24	34	11	20	7	21	42	75	16	22
3	60	402	0.35	27	33	13	24	1	12	41	69	19	22
4	63	845	0.60	13	22	15	23	15	26	43	71	17	21
5	61	3928	0.89	29	37	19	29	8	20	56	86	20	23
6	45	1996	1.04	14	32	17	26	6	14	37	72	21	22
7	52	6733	0.57	28	48	13	22	4	7	45	77	20	23
8	62	4048	2.03	24	43	10	24	7	10	41	77	15	24
9	52	1474	0.53	25	38	17	25	5	9	47	72	19	19
10	76	7697	0.68	31	40	19	26	5	8	55	74	18	24
Ave	59	3530	0.96	24	37	15	25	7	14	46	76	18	22

rent scheduling method (*'Cur'*) related to the number of operating rooms opened ($\sum_{rd} e_{rd}$), the number of early nurse shift duties ($\sum_{nd,v=1} nx_{ndv}$), the number of late nurse shift duties ($\sum_{nd,v=2} nx_{ndv}$), the number of night shift duties ($\sum_{nd,v=3} nx_{ndv}$) and the total number of shift duties ($\sum_{ndv} nx_{ndv}$). nx_{ndv} indicates the binary assignment of nurse n to shift v on day d.

The results reveal that the proposed methodology is able to reach high-quality feasible solutions with an average optimality gap ($\%Gap_{CGD}$) of 0.96% in a reasonable average computation time of 3530 s (CPU_{CGD}). This allows us to conclude that the CGD algorithm leads to a very satisfactory performance taking real-life data into account. Note that the standard MIP solver is not able to come up with a solution because of memory problems.

Further, Table 6 allows the comparison of the solution quality obtained by the CGD algorithm solution and the current procedure (*Cur*) in the hospital under consideration. It is important to point out that both approaches were able to schedule all surgical cases. However, as a result of a better planning and scheduling of the surgical cases, the proposed CGD algorithm needs only 18 operating rooms, which is less compared to the 22 rooms required by the current approach of the Sina Hospital. In addition, the proposed approach successfully leads to a smaller number of nurse shift duties resulting from a better coordination of the operating room resources via the combined surgical case scheduling with nurse re-rostering and nurse duty assignment. In particular, the proposed approach was able to significantly reduce the number of night shift duties. Overall, we reduced the total number of assigned shift duties from 755 to 457 over the 10 weeks demonstrating the more efficient use of operating room resources as a result of the proposed approach.

6.3 Value of integration

The SCPS-NRRSA problem under study considers different types of decisions related to (1) the surgical case planning and scheduling problem (SCPS), (2) the nurse re-rostering problem (NRR) and (3) the nurse surgical assignment problem (SA). These decisions are considered simultaneously in order to optimise the utilisation of the resources (OR time, nurses) via profit maximisation and to minimise the number of nurse requirements for each shift. The latter is realised by linking the nurse re-rostering decision to the nurse surgical assignment decision such that the nurse roster is operationalised and nurses can operate different (non-overlapping) patients in different operating rooms instead of assigning nurses to one more general shift duty.

As described in the literature review, these decisions are typically not integrated and are taken in a sequential manner. In order to discern the value of the integration of these problems and to justify the definition of the problem under

Table 7 Results for the value of integration on real-life data

	Scenario 1	Scenario 2	Scenario 3	Scenario 4	Cur
$NRR - \sum_{ndv} nx_{ndv}$	45.7	57.6	49.9	59.3	–
$SA - \sum_{ndv} nx_{ndv}$	45.7	49	49.9	55.9	75.5
# Shift changes	0	0	12.3	14.3	0
$\sum_{rd} e_{rd}$	18.2	17.9	16.1	15.6	22.4
CPU_{CGD}	3530	1745	8242	5721	–

study, we test different degrees of integration and evaluate four scenarios, i.e.

- **Scenario 1: SCPS ⊎ NRR ⊎ SA**
  This scenario represents the SCPS-NRRSA problem under study described in Sect. 3 that integrates the three different problems in one stage.
- **Scenario 2: SCPS ⊎ NRR | SA**
  In the first stage, this scenario integrates the SCPS and NRR problem such that the nursing resources are taken into account when scheduling the surgical cases. This problem assigns the nurses to shift duties related to a specific operating room. In the second stage, the nurses are further assigned to specific surgical cases by solving the SA problem to minimise the number of shift duties.
- **Scenario 3: SCPS | NRR ⊎ SA**
  In the first stage, the surgical case schedule is composed without considering the schedule of the nursing resources. Only the size of the nursing workforce is taken into account. In the second stage, the NRR and SA problems are solved in an integrated manner, i.e. the originally announced nurse roster is adapted to the composed surgical case schedule and simultaneously nurses are assigned to specific surgical cases to minimise the number of shift duties.
- **Scenario 4: SCPS | NRR | SA**
  This scenario solves the three problems in a sequential manner. In the first stage, the surgical case schedule is composed. In the second stage, the nurses are re-rostered according to the composed surgical case schedule. In a third stage, the number of shift duties is minimised by assigning the nurses to specific surgical cases.

Table 7 shows the results when applying the different scenarios to the real-life data and makes the comparison with the current schedule of the hospital (Cur). These scenarios are evaluated based on the average number of nurse shift duties resulting after the nurse re-rostering step ($NRR - \sum_{ndv} nx_{ndv}$), the number of nurse shift duties resulting after the nurse surgical assignment step ($SA - \sum_{ndv} nx_{ndv}$), the number of changed nurse shift duties as a result of re-rostering (#Shift changes), the number of operating rooms opened ($\sum_{rd} e_{rd}$) and the computational effort (CPU_{CGD}). Note that when the NRR and SA problems are solved in an integrated manner, there is no difference in the number of shift duties, i.e. '$NRR - \sum_{ndv} nx_{ndv}$' is equal to '$SA - \sum_{ndv} nx_{ndv}$'. The results are averaged over the 10 real-life instances.

Table 7 shows that the integration of the SCPS problem, the NRR problem and/or the SA problem (Scenario 1, 2, 3) reduces the number of shift duties but increases the number of operating rooms used compared to a pure sequential approach (Scenario 4). This results from the fact that in the sequential approach more surgical cases are assigned to one operating room to minimise the cost of opening operating rooms and the nursing resources are not considered. The fully integrated approach (Scenario 1) leads to the smallest number of nurse shift duties. Note that both Scenario 1 and 2 dominate the current schedule (Cur) as both integrated scenarios perform better in terms of number of shift duties, number of shift changes and number of operating rooms opened.

The value of introducing the SA problem in the problem definition can be discerned when comparing the number of nurse shift duties after the nurse re-rostering step ($NRR - \sum_{ndv} nx_{ndv}$) versus the number of nurse shift duties after the nurse surgical assignment step ($SA - \sum_{ndv} nx_{ndv}$). The results for scenarios 2 and 4 show that a smaller number of nurse shift duties is installed after the nurse surgical assignment step. The SA problem considers the start times of the surgical times to perform specific nurse assignments whereas the NRR problem assigns nurses only to more general shift duties. Moreover, integrating the NRR problem and the SA problem further decreases the number of shift duties, which can be concluded by observing the number of nurse shift duties after the nurse surgical assignment step of Scenario 1 (45.7) versus Scenario 2 (49) and Scenario 3 (49.9) versus Scenario 4 (55.9).

When the NRR problem is not integrated with the SCPS problem (Scenario 3 and 4), a number of shift duties needs to be changed whereas this is not the case for Scenario 1 and 2. Hence, solving the SCPS problem in an isolated manner will lead to the re-scheduling of nurses.

7 Conclusion

Different decisions related to the allocation of resources in the operating room department are based on the expected patient demand. However, the actual patient demand may differ from this expected demand leading to the inefficient use of resources. In this paper, we aim to overcome these deficiencies by integrating the surgical case planning and scheduling problem and include the nurse re-rostering decision in order to utilise the operating room resources as efficiently as possible and maximise the operating room profit. To that purpose, nurses are assigned to specific surgical cases in order to minimise the number of shift duties in an accurate manner. We propose a mathematical formulation for the problem under study and develop a two-phase heuristic procedure thriving on column generation and a diving heuristic to drive fractional solutions to integrality. The procedure is tested on a large set of artificial problem instances that are generated in a structured and controlled manner and are validated based upon a real-life problem environment. We indicate that the proposed procedure is able to obtain (near-)optimal solutions in an acceptable time span and demonstrate the contribution of each component in the algorithm. The CGH algorithm outperforms the MIP solver for the larger instances in terms of solution quality despite the fact that the running times of the general MIP solver are significantly higher. Moreover, we conducted a real-life case study involving the Sina Hospital in Iran. The computational experiments demonstrated the more efficient utilisation of the operating room resources as a result of using the proposed approach and integrating the SCPS problem with nurse re-rostering and the nurse duty assignment problem.

Our intentions for future research are twofold. First, since we explore the operational decision phase, we aim to incorporate uncertainty (e.g. the uncertain duration or arrival of surgical cases) and propose a stochastic optimisation problem to build more robust and/or stable schedules. Second, we aim to extend the problem definition to include more resource types on top of the considered resources in this paper, i.e. the operating rooms and surgical nurses. In this perspective, we aim to include the entire surgical team (surgeons, heterogeneous nurses, anaesthetists, etc.) and special operating room equipment to better plan and schedule the surgical cases.

A General MIP formulation

We defined the general mathematical formulation for SCPS-NRRSA problem in the following:

Additional decision variables

erd 1, if OR r is open on day d; 0, otherwise

nx_{ndv} 1, if nurse n is assigned to shift v, day d; 0, otherwise

npx_{ni} 1, if nurse n is assigned to surgical case i; 0, otherwise

Sn_{nd} Duty start time of nurse n on day d

Cn_{nd} Duty finish time of nurse n on day d

$DD_{i\bar{i}}$ 1, if surgical case i and $\bar{i}$ have been performed in a same day ; 0, otherwise

$$\text{Max } Z = \sum_{itrd} Rev_i \times px_{itrd} - \sum_{r} CoR_r \times e_i - \sum_{ndv} CoV_v \times nx_{ndv} \tag{33}$$

$$\sum_{tdr} px_{itrd} \leq 1 \quad \forall i \tag{34}$$

$$\sum_{i} px_{itrd} \leq 1 \quad \forall t, r, d \tag{35}$$

$$\sum_{\bar{i}} \sum_{\bar{t}=t+1}^{\bar{t}=t+t_i} px_{\bar{i}\bar{t}rd} \leq BM(1 - px_{itrd}) \quad \forall i, t, r, d \tag{36}$$

$$\sum_{it} px_{itrd} \leq BM \times e_{rd} \quad \forall r, d \tag{37}$$

$$S_i = \sum_{trd} t \times px_{itrd} \quad \forall i, t, d \tag{38}$$

$$C_i = S_i + t_i \times \sum_{trd} px_{itrd} \quad \forall i, t, d \tag{39}$$

$$\sum_{n} npx_{ni} = nu_i \times \sum_{rdt} px_{itrd} \quad \forall i \tag{40}$$

$$Cn_{nd} \geq C_i - BM \times (1 - npx_{ni}) - BM \times (1 - \sum_{r} tpx_{itrd}) \quad \forall n, i, d \tag{41}$$

$$Sn_{nd} \leq S_i + BM \times (1 - npx_{ni}) + BM \times (1 - \sum_{r} tpx_{itrd}) \quad \forall n, i, d \tag{42}$$

$$Cn_{nd} \leq \sum_{v} Et_v^V \times nx_{ndv} \quad \forall n, d \tag{43}$$

$$Sn_{nd} \geq \sum_{v} St_v^V \times nx_{ndv} \quad \forall n, d \tag{44}$$

$$\sum_{vd} nx_{ndv} \leq MA \quad \forall n \tag{45}$$

$$\sum_{v} nx_{ndv} \leq 1 \quad \forall n, d \tag{46}$$

$$\sum_{rt} (px_{itrd} + px_{\bar{i}trd}) - 1 \leq DD_{i\bar{i}} \quad \forall i, \bar{i}, d \tag{47}$$

$$S_i \geq S_{\bar{i}} - BM \times (1 - w_{i\bar{i}}) \quad \forall i, \bar{i} \tag{48}$$

$$\begin{aligned}(S_i - C_{\bar{i}}) + BM \times P_{i\bar{i}} \\ + BM \times w_{i\bar{i}} + BM \times (2 - npx_{ni} - npx_{n\bar{i}}) \\ + BM \times (1 - DD_{i\bar{i}}) \geq 0 \quad \forall n, i, \bar{i}\end{aligned} \tag{49}$$

$$t_i \times px_{itrd} \leq \sum_{\bar{t}=t}^{\bar{t}=t+t_i} A_{s\bar{t}rd} \quad \forall s, i \in \varphi_s, t, d \tag{50}$$

$$nx_{ndv} \leq ROS_{ndv} + Prf_{ndv} \quad \forall n, v, d \tag{51}$$

$$\begin{aligned}px_{itrd}, nx_{ndv}, npx_{nu}, DD_{i\bar{i}}, w_{i\bar{i}}, P_{i\bar{i}}, ad_{id}, e_r \in \{0, 1\} \\ \forall i, \bar{i}, n, v, t, r, d\end{aligned} \tag{52}$$

$$S_i, C_i, Sn_{nd}, Cn_{nd} \geq 0 \quad \forall i \tag{53}$$

For the description of the objective function (Eq. 33) and the constraints (Eqs. 34–39), we refer to the manuscript (see Eqs. 1), 10–16). Constraint (40) assigns nurses to particular surgeries based on the nurse requirements per surgery. Constraints (15) and (16) calculate the start and completion time of each surgery. Constraints (41) and (42) define the time window during which a nurse has to be on duty. The total number of time slots is a relevant upper bound in the right-hand side of the constraint, i.e. $BM = |T|$. Based on this time window, predefined shifts are assigned to nurses (constraints (43) and (44)). Constraint (45) stipulates that a nurse cannot be assigned to more than MA shifts during the planning horizon. Constraint (46) states that a nurse cannot have more than one shift per day.

Constraint (47) checks if two surgical cases are conducted on the same day. Constraint (48) concerns the order of two surgical cases. Constraint (49) stipulates that two overlapping surgery should not be assigned to the same nurse. This constraint is only relevant when both surgeries are conducted on the same day ($DD_{i\bar{i}} = 1$). Constraint (50) implies that a surgical case can only take place according to the availability of its related surgeon. Constraint (51) imposes that nurses can only be assigned to shift duties conform to their original roster or nurse preferences. Constraints (52) and (53) state the domain of the variables.

References

Agnetis, A., Coppi, A., Corsini, M., Dellino, G., Meloni, C., & Pranzo, M. (2014). A decomposition approach for the combined master surgical schedule and surgical case assignment problems. *Health Care Management Science*, *17*(1), 49–59.

Aringhieri, R., Landa, P., Soriano, P., Tànfani, E., & Testi, A. (2015). A two level metaheuristic for the operating room scheduling and assignment problem. *Computers and Operations Research*, *54*, 21–34.

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

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

Ben-Arieh, D., Wu, C. H., Waldman, A., DeLissa, L., & Weiss, J. (2014). Optimizing surgery schedule with PICU nursing constraints. *IIE annual conference and expo, 2014* (pp. 468–475).

Blake, J. T., & Donald, J. (2002). Mount Sinai Hospital uses integer programming to allocate operating room time. *Interfaces*, *32*(2), 63–73.

Breu, F., Guggenbichler, S., & Wollmann, J. (2008). *Column generation* (Vol. 5). Berlin: Springer.

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

Cardoen, B., Demeulemeester, E., & Beliën, J. (2010). Operating room planning and scheduling: A literature review. *European Journal of Operational Research*, *201*(3), 921–932.

Chaabane, S., Meskens, N., Guinet, A., & Laurent, M. (2008). Comparison of two methods of operating theatre planning: Application in Belgian Hospital. *Journal of Systems Science and Systems Engineering*, *17*(2), 171–186.

Day, R., Garfinkel, R., & Thompson, S. (2012). Integrated block sharing: A win–win strategy for hospitals and surgeons. *Manufacturing and Service Operations Management*, *14*(4), 567–583.

Di Martinelly, C., & Meskens, N. (2017). A bi-objective integrated approach to building surgical teams and nurse schedule rosters to maximise surgical team affinities and minimise nurses' idle time. *International Journal of Production Economics*, *191*(May), 323–334.

Di Martinelly, C., Baptiste, P., & Maknoon, M. (2014). An assessment of the integration of nurse timetable changes with operating room planning and scheduling. *International Journal of Production Research*, *52*(24), 7239–7250.

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

Essen, J. T. V., Hans, E. W., Hurink, J. L., & Oversberg, A. (2012). Minimizing the waiting time for emergency surgery. *Operations Research for Health Care*, *1*(2–3), 34–44.

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

Fei, H., Meskens, N., Combes, C., & Chu, C. (2009b). The endoscopy scheduling problem: A case study with two specialised operating rooms. *International Journal of Production Economics*, *120*(2), 452–462.

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

Fügener, A., Hans, E. W., Kolisch, R., Kortbeek, N., & Vanberkel, P. T. (2014). Master surgery scheduling with consideration of multiple downstream units. *European Journal of Operational Research*, *239*(1), 227–236.

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

Guinet, A., & Chaabane, S. (2003). Operating theatre planning. *International Journal of Production Economics*, *85*(1), 69–81.

Guo, M., Wu, S., Li, B., & Rong, Y. (2014). Maximizing the efficiency of use of nurses under uncertain surgery durations: A case study. *Computers and Industrial Engineering*, *78*, 313–319.

Gür, A. (2018). Application of operational research techniques in operating room scheduling problems: Literature overview. *Journal of Healthcare Engineering*,. https://doi.org/10.1155/2018/5341394.

Hadj, A. B., Ladet, P., Jebali, A., Hadj Alouane, A. B., & Ladet, P. (2006). Operating rooms scheduling. *International Journal of Production Economics*, *99*(1–2), 52–62.

Joncour, C., Michel, S., Sadykov, R., Sverdlov, D., & Vanderbeck, F. (2010). Column generation based primal heuristics. *Electronic Notes in Discrete Mathematics*, *36*, 695–702.

Lamiri, M., Xie, X., & Zhang, S. (2008). Column generation approach to operating theater planning with elective and emergency patients. *IIE Transactions*, *40*(9), 838–852.

Latorre-Núñez, G., Lüer-Villagra, A., Marianov, V., Obreque, C., Ramis, F., & Neriz, L. (2016). Scheduling operating rooms with consideration of all resources, post anesthesia beds and emergency surgeries. *Computers and Industrial Engineering*, *97*, 248–257.

Lim, G. J., Mobasher, A., Bard, J. F., & Najjarbashi, A. (2016). Nurse scheduling with lunch break assignments in operating suites. *Operations Research for Health Care*, *10*, 35–48.

Lübbecke, M. E., & Desrosiers, J. (2005). Selected topics in column generation. *Operations Research*, *53*(6), 1007–1023.

Maenhout, B., & Vanhoucke, M. (2011). An evolutionary approach for the nurse rerostering problem. *Computer and Operations Research*, *38*, 1400–1411.

Magerlein, J. M., & Martin, J. B. (1978). Surgical demand scheduling: A review. *Health Services Research*, *13*(4), 418–433.

Marques, I., Captivo, M. E., & Vaz Pato, M. (2014). A bicriteria heuristic for an elective surgery scheduling problem. *Health Care Management Science*.

Marynissen, J., & Demeulemeester, E. (2016). Literature review on integrated hospital scheduling problems. FEB Research Report.

Meskens, N., Duvivier, D., & Hanset, A. (2013). Multi-objective operating room scheduling considering desiderata of the surgical team. *Decision Support Systems*, *55*(2), 650–659.

Mobasher, A., Lim, G., Bard, J. F., & Jordan, V. (2011). Daily scheduling of nurses in operating suites. *IIE Transactions on Healthcare Systems Engineering*, *1*(4), 232–246.

Monteiro, T., Meskens, N., & Wang, T. (2015). Surgical scheduling with antagonistic human resource objectives. *International Journal of Production Research*, *53*(24), 7434–7449.

Moz, M., & Pato, M. (2003). An integer multicommodity flow model applied to the rerostering of nurse schedules. *Annals of Operations Research*, *119*, 285–301.

Ogulata, S. N., & Erol, R. (2003). A hierarchical multiple criteria mathematical programming approach for scheduling general surgery operations in large hospitals. *Journal of Medical Systems*, *27*(3), 259–270.

Ozkarahan, I. (1995). Allocation of surgical procedures to operating rooms. *Journal of Medical Systems*, *19*(4), 333–352.

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

Riise, A., Mannino, C., & Burke, E. K. (2016). Modelling and solving generalised operational surgery scheduling problems. *Computers and Operations Research*, *66*, 1–11.

Roland, B., & Riane, F. (2011). Integrating surgeons' preferences in the operating theatre planning. *European Journal of Industrial Engineering*, *5*(2), 232.

Roland, B., Di Martinelly, C., & Riane, F. (2007). Operating theatre optimization: A resource-constrained based solving approach. In *Proceedings—ICSSSM'06: 2006 international conference on service systems and service management* (Vol. 1, pp. 443–448)

Roland, B., Di Martinelly, C., Riane, F., & Pochet, Y. (2010). Scheduling an operating theatre under human resource constraints. *Computers and Industrial Engineering*, *58*(2), 212–220.

Sadykov, R., Vanderbeck, F., Pessoa, A., & Tahiri, I. (2019). Primal heuristics for branch-and-price: The assets of diving methods. *INFORMS Journal on Computing*, *31*, 251–267.

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

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

Souki, M., & Rebai, A. (2012). Heuristics for the operating theatre planning and scheduling. *Journal of Decision Systems*, *19*, 37–41.

van Essen, J. T., Hans, E. W., Hurink, J. L., & Oversberg, A. (2012). Minimizing the waiting time for emergency surgery. *Operations Research for Health Care*, *2–3*(6), 34–44.

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

Velasquez, R., & Melo, M. (2006). A set packing approach for scheduling elective surgical procedures. In H.-D. Haasis, H. Kopfer, & J. Schönberger (Eds.), *Operations research proceedings 2005: Selected Papers of the Annual International Conference of the German Operations Research Society (GOR), Bremen, September 7–9, 2005* (pp. 425–430). Berlin: Springer.

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

Villarreal, M. C., & Keskinocak, P. (2016). Staff planning for operating rooms with different surgical services lines. *Health Care Management Science*, *19*(2), 144–169.

Wang S, Ma C, Xiang W (2015) Operating rooms decision optimization integrating surgery planning and nurse rostering. In *Lecture notes in electrical engineering, lecture notes in electrical engineering* (Vol. 349, pp. 919–926). Berlin: Springer.

Wang, Y., Tang, J., & Fung, R. Y. (2014). A column-generation-based heuristic algorithm for solving operating theater planning problem under stochastic demand and surgery cancellation risk. *International Journal of Production Economics*, *158*, 28–36. https://doi.org/10.1016/j.ijpe.2014.07.015.

Xiang, W., Yin, J., & Lim, G. (2014). A short-term operating room surgery scheduling problem integrating nurse constraintspdf. *Artificial Intelligence in Medicine*, *63*(2), 91–106.

Zhang, Z., & Xie, X. (2015). Simulation-based optimization for surgery appointment scheduling of multiple operating rooms. *IIE Transactions*, *47*(9), 998–1012.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.