



Scheduling of elective operations with coordinated utilization of hospital beds and operating rooms

Zhaohui Li¹ · Haiyue Yu²  · Zhaowei Zhou³

Accepted: 12 April 2024 / Published online: 18 May 2024

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2024

Abstract

The problem studied in this paper is elective surgery scheduling, with resource constraints in each of the three following stages: preoperative, perioperative, and postoperative stages. With the integrated availability of hospital beds in wards and operating rooms, the aim is to determine operation start times of surgeries and allocate the hospital beds to patients while getting patients treated as soon as possible. This task is crucial in providing timely treatments for the patients while ensuring the hospital's resource utilization balance. For the problem, we first formulate it as mixed-integer programming, which is NP-complete. Then, we propose several heuristics to overcome the long computation time. To make the solution better, we also propose improved algorithms. Finally, we conduct a series of numerical studies to illustrate the efficiency of our proposed algorithms and examine the impact of the number of jobs, beds, and surgery blocks on the performance measure. Computational experiments showed the superior performance of our heuristics in makespan.

Keywords Surgery scheduling · Bed assignment · Operating room · Hospital bed · Parallel machine

1 Introduction

Hospitals are vital components of the healthcare sector worldwide, with the operating room (OR) as a critical hub for both costs and revenue (Nasiri and Rahvar 2017; Denton et al. 2007). Enhancing OR efficiency, even slightly, can boost patient survival

✉ Haiyue Yu
haiyueyu2022@163.com

¹ School of Management and Economics, University of Electronic Science and Technology of China, Chengdu, China

² Shanghai Municipal Hospital of Traditional Chinese Medicine, Shanghai University of Traditional Chinese Medicine, Shanghai, China

³ Ministry of Education Key Laboratory of Metabolism and Molecular Medicine, Department of Endocrinology and Metabolism, Zhongshan Hospital, Fudan University, Shanghai, China

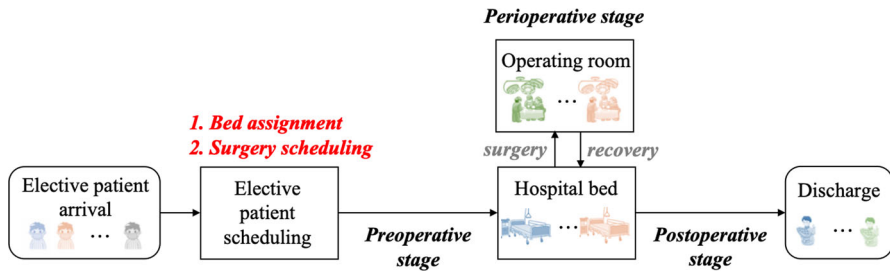


Fig. 1 The description of the procedure

rates, improve services, and enhance stakeholder satisfaction, including patients, surgeons, and OR staff (Macario et al. 1995; Hamid et al. 2019). This, in turn, contributes to a hospital's income. Therefore, optimizing OR efficiency is of utmost importance. Additionally, hospital beds play a crucial role for patients, and a shortage of beds can lead to delayed surgeries or premature patient discharge Lane et al. (2000). Inefficient bed assignments further contribute to the suboptimal utilization of OR resources, intensifying the challenges faced by healthcare facilities. Consequently, the joint decision-making process regarding surgery scheduling and bed allocation emerges as a pressing and intricate issue in healthcare administration.

Surgical patients are grouped into emergency, urgent, and elective classes Pham and Klinkert (2008). Emergency cases must be performed within two hours of the decision to operate, whereas urgent ones must be within a few hours Guerriero and Guido (2011). Elective cases that occur frequently can be planned in advance and usually performed to suit both patient and surgeon. Thus, hospitals can increase the efficiency of the operating room by assigning these elective patients to the most appropriate period. Our goal is to assist decision-makers with elective patient scheduling.

Specifically, in a typically surgical path, each patient goes through three stages: pre-operation (surgery preparation), surgery, and recovery (Erdogan et al. 2011; Saremi et al. 2013; Erdogan et al. 2015). In the first stage (*preoperative stage*), patients need to be assigned to the *hospital wards* in advance. An OR nurse identifies and extracts the patient's charts and information, such as lab results and consent forms. This stage includes different preparation procedures required for each type of surgery, such as taking drugs and anesthetics, having blood tests, and waiting for the medicine to take effect. The surgery stage (*perioperative stage*) includes anesthesia and operation. The last stage (*postoperative stage*) accounts for the procedures and the time required for the patient's recovery. After surgery, most patients are taken to the postanesthetic care unit (PACU), where they recover from anesthesia's residual effects under PACU nurses' care. After their stay in PACU, inpatients usually return to *their wards* before being discharged. Figure 1 represents the three stages of a surgical path. In this sense, the hospital bed is an essential resource for elective patients. Most studies on elective surgery scheduling problems in the literature focus on utilizing resources directly related to surgery, such as physicians, nurses, and surgical equipment (Xiang et al. 2015; Behmanesh and Zandieh 2019; Yu et al. 2022). This paper solves the problem of incorporating OR scheduling and bed assignment decisions.

Some works consider preoperative activities (e.g., blood sampling and anesthetic consult before surgery) and resources after surgery such as recovery, PACU, and ICU beds into surgery scheduling problems (Guido and Conforti 2017; Behmanesh and Zandieh 2019; Moosavi and Ebrahimnejad 2020; Makboul et al. 2022; Santos and Marques 2022). The operating room scheduling problem are mainly discussed at the operational level while the bed resources are mainly studied at the tactical level. However, in practical applications, there is a need to simultaneously make tactical-level decisions regarding operating room scheduling and bed resource allocation. The latter involves determining specific admission times for patients and the beds to which they will be assigned. If operational-level decision-making only focuses on operating room planning issues, the utilization of bed resources may not have reached optimal efficiency yet. Motivated by the disparity between reality and existing literature, this paper explores the optimization problem of jointly making surgical scheduling and bed allocation decisions at the operational level.

In this paper, we adopt the makespan, the completion time of the last work, to measure coordinated utilization of hospital beds and operating rooms, which has practical meaning. For example, during the outbreak of COVID-19, elective surgeries were delayed or temporarily halted to mitigate the risk of virus transmission and put anti-virus activities first. Hospitals need to balance case priority, patients' safety, and precious healthcare resources such as surgeons, staff, personal protective equipment (PPE), and other essential equipment (e.g., ventilators). According to Meredith et al. (2020), a survey by Massachusetts General Hospital shows that, from May to July 2020, 31.7% of patients with breast cancer reported delayed experience in screening or treatment. After massive delays of elective surgeries, it is highly urgent to recover regular treatment for patients influenced by delays as soon as possible. In other words, the accumulation of surgeries needs to be finished as quickly as possible. In this sense, we optimize the makespan to maximize the coordinated utilization of hospital beds and operating rooms that are responsive to actual needs.

We make two decisions to optimize the makespan, assigning the patients to each bed and scheduling the surgery date for patients on each bed. Because this problem involves parallel machine scheduling problems, it is NP-complete and computationally intractable to solve to optimal. Due to the difficulties of solving our studied scheduling problem of elective operations with coordinated utilization of hospital beds and operating rooms, we try to find heuristic solutions for practical concern. First, we develop a mixed-integer programming formulation of the $Pm|restw|Cmax$ problem. Then, we propose five simple heuristics based on the Longest Processing Time (LPT) rule, workload balance, and construct the composite index. Finally, we put forward improved MIP-based heuristics with the solutions of the simple heuristics that we proposed as initial solutions. The experimental results show that the heuristics are practical and efficient.

We make the following contributions to the surgery scheduling literature. First, compared with the current work on surgery scheduling, we extend the work by simultaneously considering the utilization of hospital beds and operating rooms. To the best of our knowledge, we are the first to study the joint OR scheduling and bed assignment decisions at the operational level. Second, from a methodological point of view, we develop practical heuristics to achieve a near-optimal solution.

This paper proceeds as follows. Section 2 reviews related works. Section 3 gives a general problem statement and proposes a mixed-integer linear programming model of the problem. Section 4 develops several heuristics to solve the original problem. Section 5 evaluates the numerical performance of the proposed algorithms with various instances. Finally, Sect. 6 concludes the paper and discusses future studies.

2 Literature review

Given that our work combines different problems, including scheduling beds and operation rooms, our study can relate to several literature streams. We present the literature review in two parts: surgery scheduling and patient admission problems, corresponding to the main issues involved.

Surgery scheduling is a classic problem for operation room scheduling that has attracted wide attention in the literature. According to Beliën and Demeulemeester (2007) and Hulshof et al. (2012), there are three-stage decisions involved in master surgery planning: strategic, tactical, and operational. The location, size, and capacity of operation rooms are decided strategically. This first stage is also called case-mix planning since it determines which ailment's capacity will be preserved. Several linear programming models have been proposed for case-mix planning (e.g., Hughes and Soliman 1985; Blake and Carter 2002), and Hof et al. (2017) provide the first literature review focusing on case-mix planning problems. At the tactical level, the provider mainly assigns surgery blocks, staff, and resources to physicians, which involves the development of a master surgery schedule (MSS). This stage has a wide range of approaches in the literature. Testi et al. (2007), Vanberkel et al. (2011), Mannino et al. (2012), and more recently, Penn et al. (2017) use exact methods. At the same time, heuristic approaches (Beliën and Demeulemeester 2007; Hosseini and Taaffe 2015; Marchesi and Pacheco 2016; Dellaert and Jeunet 2017) or simulation models Cappanera et al. (2014) are also proposed. At the operational level, the hospital arranges specific interventions for each patient. We can further divide the operational level into two main stages: the advanced scheduling of assigning patients to surgery blocks and the allocation scheduling of sequencing the surgeries in surgery blocks.

Our problem can be located in the advanced scheduling stage at the operational level. Inputs to the problem are data about departments, rooms, patients, and MSS resulting from tactical planning, and the output is the assignment of patients to rooms, OR slots, and admission days. As for this problem, some papers consider preoperative, intraoperative, and postoperative stages similar to ours. For example, Xiang et al. (2015) proposed an Ant Colony Optimization (ACO) method to efficiently solve the problem based on the knowledge gained in the Flexible Job Shop Scheduling Problem (FJSSP) by observing the similarities between operating room surgery scheduling, and multi-resource constrained FJSSP in manufacturing. Guido and Conforti (2017) proposed a multi-objective integer linear programming model to efficiently plan and manage hospital operating room suites and exploit a novel hybrid genetic solution approach. Belkhamza et al. (2018) aimed to minimize the maximum end time of the last activity in the postoperative stage and the total idle time in the operating room and provided two meta-heuristics (Iterative Local Search and Hybrid Genetic

Algorithm) to schedule surgeries. Behmanesh and Zandieh (2019) addressed the bi-objective surgical case scheduling problem with uncertain service times by developing a novel bi-objective ant system (Fuzzy Pareto Envelope-based Selection Ant System). Yu et al. (2022) developed an improved multi-objective imperialist competitive algorithm (IMOICA) to optimize the whole scheduling considering patients' switching and preparation time in different stages. Although they also incorporate the preoperative, intraoperative, and postoperative stages in the model, they treat the resources in the preoperative and postoperative stages separately, including nurses, recovery, PACU, and ICU beds. Unlike them, we mainly consider one type of resource shared by both preoperative and postoperative stages, the hospital beds. Each patient occupies a bed throughout their entire hospital stay from the admission day to the discharging day.

The patient admission scheduling Problem (PASP) is well-reported in the research literature. PAS assigns patients to beds to fulfill their medical concerns and personal wishes as much as possible. This process can be carried out at three levels: strategic, tactical, and operational Demeester et al. (2010), similar to surgery scheduling. At the strategic level, the hospital wants to make long-term structural decisions to maximize organizational efficiency. They often use historical patient admissions data to estimate future needs and determine the appropriate resources required (e.g., available beds) (Harper 2002; Vissers et al. 2007; Chen et al. 2010). At the tactical level, there are seasonal differences in demands and multiple available resources. The admission policies are specified based on the number of necessary resources, the combination of inpatients, and guidelines of operational planning (Chen et al. 2010; Holm et al. 2013; Barz and Rajaram 2015). Most studies assume that the hospital has only one potential bottleneck. Some papers consider more than one constraining resource, including operation rooms and beds. They decide whether to accept, postpone, or reject the admission from a deterministic model (Adan and Vissers 2002; Vissers et al. 2005) or random stream of non-emergency elective patient Barz and Rajaram (2015). Our paper focuses on the accepted patients, assigns patients to different beds and decides the surgery day for each patient. Finally, the operational level decision defines scheduling rules governing daily patient admission planning. The class PASP in the operational level is a static offline problem that assumes deterministic demand requirements and fixed admission dates and length-of-stay (Demeester et al. 2010; Bastos et al. 2019). Ceschia and Schaerf (2011), Ceschia and Schaerf (2016) and Vancroonenburg et al. (2012) propose dynamic problems in which admission and discharging dates are not known in advance. Zhu et al. (2020) develop a Markov Decision Process (MDP) model to decide how many elective patients the hospital should admit each day to use both operating room and inpatient bed capacity optimally. Our problem is similar to the issues from the perspective of bed assignment, but we add decisions about surgery scheduling.

Some studies have concurrently addressed the decision-making processes related to surgery scheduling and bed allocation (Guido and Conforti 2017; Behmanesh and Zandieh 2019; Moosavi and Ebrahimnejad 2020; Makboul et al. 2022; Santos and Marques 2022). As an illustrative instance, Makboul et al. (2022) devised the Master Surgical Schedule (MSS) within a one-week timeframe, taking into account constraints within the operating theater (OT), patient preferences, and factors in the availability of both intensive care unit (ICU) beds and post-surgery beds. The work of

Moosavi and Ebrahimnejad (2020) delved into the operational room planning problem at both tactical and operational levels, considering upstream and downstream units. It is worth noting that they deal with the bed resource at the tactical level, and the operating rooms scheduling at the operational level. In contrast, our research adopts a distinct approach. In this paper, we explore the intertwined complexities associated with surgery scheduling and bed allocation, addressing both dimensions within the framework of operational decision-making.

Moreover, the combinatorial nature of our surgery scheduling problem makes it computationally difficult to solve it optimally. Note that as an interesting comparison, Xu et al. (2008) studied a parallel machine with periodic maintenance to minimize the makespan, who also related their problem to combined $Pm|Cmax$ with the Bin-packing problem. There exist two significant differences between their work and ours. In their study, maintenance represents periodic resource unavailability in terms of resources. Therefore, they mainly deal with one resource, namely the machine. We deal with two sorts of resources, hospital beds and operating rooms. Moreover, in their problem, a bin denotes the time interval between consecutive maintenance, while in ours, a bin denotes the time interval of the surgery block, and patients from different beds will be packed in one surgery block. The bin-packing decision of surgery scheduling affects all beds and therefore is more complex. Zhong et al. (2014), Wang et al. (2015), Su et al. (2017) model surgery scheduling as a deterministic parallel machine with resource constraints and machine eligibility and develop heuristics to deal with different objectives. We adopt similar methodologies to their studies, but the major concerns and job characteristics are significantly different. They mainly focus on the scheduling of the operating room. However, we study a surgery scheduling problem with coordinated utilization of hospital beds and operating rooms, considering the bed as a resource that lasts the entire surgical process.

3 Problem formulation

3.1 Problem description

This section describes our problem and proposes Mixed Integer Programming with a deterministic daily-based discrete-time formulation. As described in Sect. 1, in a typical surgical path, each patient goes through three stages: In the first stage (pre-operative stage), patients need to be assigned to the hospital wards in advance for preparation procedures. The surgery stage (perioperative stage) includes anesthesia and operation. After surgery (postoperative stage), patients return to their beds for recovery. In this study, we mainly focus on bed assignment and surgery scheduling at the operational level within a certain department. We operate under the assumption that the surgery blocks and the corresponding operating rooms can serve the patients under consideration indiscriminately. Given the data about the available surgery blocks and corresponding operating rooms from the tactical planning and the available number of beds from the strategic planning, the decisions include the admission days, the assignment of patients to beds and the allocation of operating room slots within a specific department. The objective is to maximize the utilization of hospital beds and surgery

Table 1 Processing times of 2 types

	p(days)	s(units)	q(days)
Type 1	1	1	1
Type 2	2	2	2

rooms, measured by minimizing the makespan to treat the current patients as soon as possible.

Following the three-field notation in scheduling theory, our problem can be categorized as $Pm|res, tw|Cmax$, where res indicates the resources have capacity constraints. tw denotes the time window, indicating there are exogenous time intervals when resources are available. The surgery resource, or surgery duration in our problem is measured by time units, corresponding to OR slots, and is capacitated by the length of the block or the regular working hours. Patients that receive surgeries in the same day share the surgical resources of that day. The preoperative duration and postoperative duration are measured by days that a patient stays in bed before and after surgery.

The problem is modeled following assumptions often made in the literature (Jebali et al. 2006; Testi and Tànani 2009; Fei et al. 2010; Marques et al. 2014; Castro and Marques 2015; Roshanaei et al. 2020):

- The preoperative, postoperative, and surgical durations are deterministic and can be estimated by historical data.
- ORs and beds are homogeneous.
- ORs capacity is fixed.
- A hospital bed can be assigned with at most one patient at a day.
- A patient occupies a hospital bed during his/her stay duration.
- The patient will be admitted to a bed at the beginning of a day, and he/she will leave at the end of a day once recovered from surgery.

3.2 An example of the objective

Traditional surgery room scheduling mainly focuses on maximizing the utilization of the operating rooms, which is indeed an important and expensive resource. We argue that bed is also a scarce resource featuring long time occupancy and prerequisite of specific surgery. We use an example to illustrate the importance of coordinated scheduling of beds and operating rooms.

Suppose one hospital department is facilitated with 1 operating room with 2 time units of surgery capacity each day and 4 beds. There are 2 types of patients. Table 1 describes the processing time durations of the two types of patients, where p is the pre-surgery time duration, s is the surgery duration, and q is the post-surgery time duration. There are 8 patients waiting to be treated. Table 2 shows the type of each patient.

To treat all of the 8 patients, 12 units of surgery resources are required. To make full use of surgical resources, it is optimal to schedule the 8 surgeries in 6 full days. In other words, any schedule with 6-day surgery should be indifferent to merely maximize

Table 2 Type of each patient

Patient No	1	2	3	4	5	6	7	8
Type	2	2	2	2	1	1	1	1

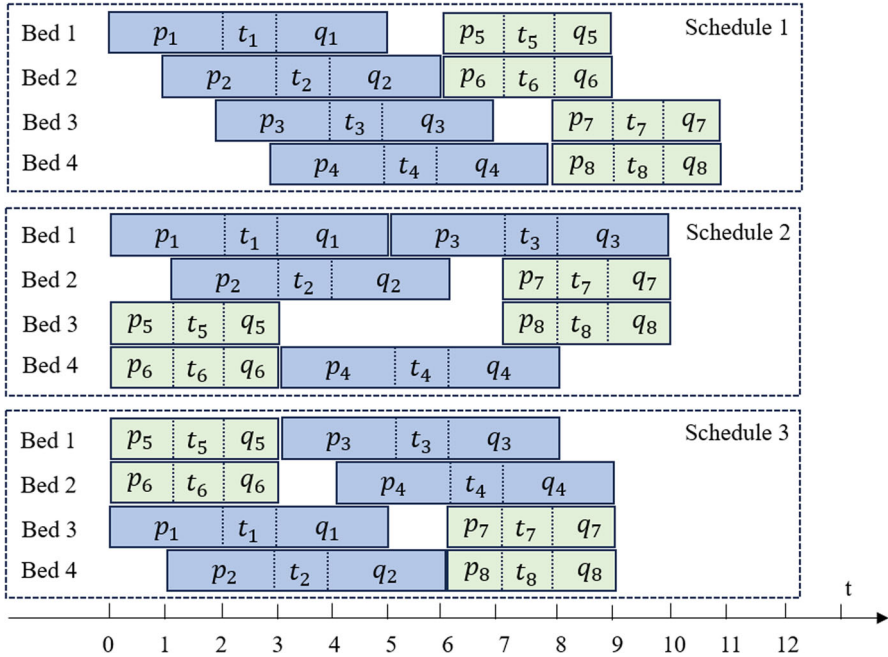


Fig. 2 3 schedules with the same utilization of the operating room

operating room utilization. In Fig. 2, we display three schedules using Gantt charts. The horizontal axis t represents the time horizon (days). $t_j, j = 1, \dots, 8$ represent the surgery day of each patient. $p_j + q_j + 1, j = 1, \dots, 8$ denote the total duration that each patient stays in the hospital. The three schedules in Fig. 2 all satisfy this optimal utilization of operation rooms. If we also consider bed utilization, the three schedules are significantly different in performance. Since patients require different pre-surgery tests, surgery durations and post-surgery recovery durations, there might be idleness of beds due to the operation capacity constraints. Schedule 3 treats and releases all patients on day 9 while others last longer. Thus, Schedule 3 makes better use of beds. The difference in bed utilization is reflected by the different makespans. However, due to the limited number of beds, an improper operating room schedule will also impact the days scheduled to perform surgeries, and further impact the admission and discharge schedules of patients, leading to variations in makespan. As a result, we adopt makespan as the measure to reflect the coordinated utilization of hospital beds and operating rooms. In general, the shorter the makespan is, the more balanced the utilization of beds and operation rooms.

When the two types of resources, namely beds and operating rooms, are highly unbalanced, one resource becomes a bottleneck, and the other is seldom fully utilized.

Table 3 Notation

Notation	Explanation
Parameters	
I	Set of patient types
J	Set of patients
K	Set of beds
T	Set of time horizon (days)
p_j	Pre-surgery processing duration of patient j (days)
s_j	Surgery duration of patient j (units)
t_j	Surgery day that patient j is scheduled to receive operation
q_j	Post-surgery processing duration of patient j (days)
c_t	Surgery resource capacity at day t (units)
M	A sufficient large number

Therefore, coordinated scheduling reduces to maximizing single resource utilization, such as the well explored surgery scheduling in the literature.

To avoid trivial scenarios, we identify two settings where coordination fails to improve the system, which our study will not address. The notations are listed in Table 3. If a surgical team is assigned with periodic surgery block of length l , they perform surgeries every other l day. We have the following results.

Proposition 1 *When there are $|J|$ patients with different $p_j, s_j, q_j, \forall j \in J, |K|$ beds and a periodic surgery cycle of length l and capacity c .*

- *If $l \geq \arg \max_j p_j + q_j + 1$, the beds cannot realize full utilization.*
- *If $c \geq \arg \max_j |K|s_j$, the operating rooms cannot realize full utilization.*

The insight behind the two scenarios is straightforward. Therefore we omit the proof. For the rest analysis, we consider the setting when $l < \arg \max_j p_j + q_j + 1$, $\frac{c}{|K|} < \arg \max_j s_j$, and two types of resources are relatively balanced.

3.3 Mixed integer programming

We first define the decision variables of our studied problem and develop a Mixed Integer Programming. To describe the programming, we supplement new notations in Table 4.

Table 4 New notations

Notation	Explanation
Decision variable	
x_{jk}^t	=1, if patient j is admitted to bed k at day $t, \forall j \in J, k \in K, t \in T$; =0, otherwise
y_j^t	=1, if patient j is performed surgery at day $t, \forall j \in J, t \in T$; =0, otherwise
C_j	Discharging day of patient j
C_{max}	Maximum discharging day of all patients in J

$$\begin{aligned}
 \text{(P1)} \quad & \min C_{max} && (1) \\
 \text{s.t.} \quad & C_{max} \geq C_j, && \forall j \in J; \quad (2) \\
 & \sum_t \sum_k x_{jk}^t = 1, && \forall j \in J; \quad (3) \\
 & \sum_j x_{jk}^t \leq 1, && \forall k \in K, t \in T; \quad (4) \\
 & \sum_{i \neq j} \sum_{t'=t}^{t+p_j+q_j} x_{ik}^{t'} \leq M(1 - x_{jk}^t), && \forall j \in J, k \in K, t \in T; \quad (5) \\
 & \sum_k x_{jk}^t = y_j^{t+p_j}, && \forall j \in J, t \in T; \quad (6) \\
 & \sum_j y_j^t s_j \leq c_t, && \forall t \in T; \quad (7) \\
 & C_j \geq t y_j^t + q_j, && \forall j \in J, t \in T; \quad (8) \\
 & x_{jk}^t \in \{0, 1\}, && \forall j \in J, k \in K, t \in T; \quad (9) \\
 & y_j^t \in \{0, 1\}, && \forall j \in J, t \in T; \quad (10) \\
 & C_j \geq 0, && \forall j \in J \quad (11)
 \end{aligned}$$

Objective (1) is to minimize the makespan of the coordinated schedule of beds and operating rooms, which can be interpreted as maximization of utilization. Constraints (2) explain that the makespan is the maximum discharging day of all patients. Constraints (3) ensure that each patient will be allocated to one bed once in the time horizon. Constraints (4) indicate that at any time, a bed can hold at most one patient. Constraints (5) represent that a bed is occupied by a patient throughout the pre-surgery time duration, surgery day, and post-surgery recovery duration. Constraints (6) imply that the surgery will be operated after the patient undergoes a pre-surgery test. Constraints (7) restrict the number of surgeries performed in one day to below capacity, the maximum working hours. Constraints (8) represent that a patient will leave the hospital and release the bed he/she occupied after post-surgery recovery.

Lemma 1 *This problem is NP-complete.*

Table 5 New notations for heuristics

Notation	Explanation
r_t	Surgical resources that remained on the day t
R_k	Release day of bed k , which means bed k can receive new patients on the day R_k
S_j	Surgery day of the patient j
α_i	Composition of populations of patients with type i
β_i	Composition of patients in terms of surgery duration, $\beta_i = \alpha_i s_i$
γ_i	Normalized percentage of resource consumption of each type, $\gamma_i = \frac{\beta_i}{\sum_i \beta_i}$
a_t^k	Forward time availability for bed k and surgery day t
b_t^k	Afterward time availability for bed k and surgery day t

Proof We combine the bed occupancy and surgery scheduling in one model. Given bed occupancy, $Pm|Cmax$ for an arbitrary number of machines is NP-complete and is a special case of our problem. Surgery scheduling in our setting is a generalized Bin-packing problem that is also NP-complete. Our problem involves the interaction of two types of decisions and therefore is NP-complete. \square

4 Heuristics

There are two major types of resources in our study, namely bed and operating room. We need to coordinate the utilization of both resources in order to speed up throughput. As explained in the previous section, the problem is NP-complete and time-consuming to solve to optimal. Moreover, the solution obtained by the solver can be hard to interpret in actual practice. Therefore, we introduce five heuristics: fast to achieve a good feasible solution and the idea behind which is relatively intuitive. To describe the heuristics, we supplement some new notations in Table 5.

Note that LPT is one of the most commonly adopted dispatching rules that whenever a machine becomes available for assignment, assign the job with the largest processing time among those jobs not yet assigned Braglia and Petroni (1999). LPT works well in makespan minimization in parallel machine problem and has proven performance bound $(\frac{4}{3} - \frac{1}{3m})$ Michael (2018). A natural approach is to modify the LPT rule for our study. A key difference in our problem is that there can be idleness between consecutive patients in the same bed because the given surgical blocks are fixed. As a result, a simple LPT rule does not guarantee a feasible schedule. As we have two resources, in our heuristics, We focus on scheduling one resource and maintaining the feasibility of the other along the path with of idea of LPT involved.

4.1 LPT-RES heuristic

Our first heuristic is a modified LPT heuristic. The idea of modification is to assign patients to surgical resources instead of beds. For each surgical day, assign patients with the longest stay duration to OR slots first until the surgical resource exhausts.

Algorithm 1 LPT-RES Heuristic

```

Re-index patients in LPT order of  $p_j + q_j$  into unscheduled list  $J^c$ .
Initialize  $R_k = 0$ ,  $\forall k \in K$  and  $r_t = c_t$ ,  $\forall t \in T$ .
for Each surgical day  $t$  with surgical resource  $c_t$ , do
  Sort beds into set  $K_t$  in increasing  $R_k$ .
  for  $k$  in  $K_t$  do
    If  $R_k \leq t$ ,  $c_t \geq 0$ , and  $J^c$  is not empty
      for  $j$  in  $J^c$  do
        If  $r_t \geq s_j$  and  $R_k + p_j \leq t$ , set  $S_j = t$ ,  $R_k = t + q_j + 1$ . Remove  $j$  from  $J^c$ .
        Otherwise, set  $j = j + 1$ 
      end for
    end for
  end for
end for

```

First, we sort patients with the LPT rule of $p_j + q_j$. Then we consider each surgery date to assign patients to a bed that is first available. As we keep track of bed release dates, the bed is available to admit a patient for a specific surgery date only if there is enough surgical resource for the patient, the interval between admission day and surgery date is enough for preoperative tests. It is easy to see that the heuristic takes $O(|T||K||J|)$ time to complete the scheduling.

The computational investigation, which is illustrated in Sect. 5, shows the performance of LPT is unstable. This probably results from the surgical resource assignment being a generalized Bin-packing problem, also a NP-hard problem. The restriction of surgical resources creates idleness and leads to poor utilization of both resources, which is the primary reason prohibiting the LPT rule from performing effectively.

4.2 LPT-WB heuristic

When there are several types of patients, and the composition of different types is relatively steady, it is convenient and practical to analyze a typical pattern to balance workload with insight from a simple dispatching rule. We define some notations as we describe the heuristic.

The types of patients and the percentage of different types are usually exogenous, relatively steady in the short-term, and slowly evolving as time flows. Based on historical data, we use i to indicate type i where patients from the same type have the same p_i , q_i and s_i . We assume the types are arranged in the decreasing order of the bed durations. $\{\alpha_i\}_{i=1}^I$ to denote the composition of populations of patients with different types such that the proportion of type i patient is $\frac{\alpha_i}{\sum_i \alpha_i}$. And $\{\beta_i = \alpha_i s_i\}_{i=1}^I$ roughly represents composition of patients in terms of surgery duration. This is the average operating room resource consumption of different types in a steady state. The basic idea is that given the number of patients if we assign the same percentage of patients of each type for each operation day, the workload should be overall balanced.

The type with a shorter bed duration is more flexible with bed occupancy considered, as indicated by the LPT rule. We propose a protection level to provide priority for types with longer bed duration. We define $\{\gamma_i = \frac{\beta_i}{\sum_i \beta_i}\}_{i=1}^I$ as the pattern of patients, which represents the normalized percentage of resource consumption of each type.

We introduce the cumulative threshold $\{CR_i = \sum_{i' \leq i} \gamma_{i'}\}_{i=1}^I$ for type i . The surgical resources occupied by type i , denoted as A_i . When the cumulative surgical resources occupied by types prior to i , $\sum_{i' \leq i} A_{i'}$, exceeds the threshold CR_i , we will not assign the patient to balance workload of different types of patients.

If the decision is made on the rolling horizon, the pattern can be set steady in the short term and gradually adjusted by observing patients arriving data in history. If it is a single launch to get the patient pool cleared as soon as possible, patterns are easily modified during implementation. After the pattern is identified, we need to schedule a bed and operating room. At each time operating room is open for surgeries, following the pattern, sorting beds concerning available accommodation time before operation day, and assigning patients from type 0 to type I . Upon each assignment, four conditions should be satisfied. (1) There are type i patients not scheduled yet. (2) There is enough surgical resource for type i patients. (3) There exists an empty bed available for accommodation type i . (4) Surgical resource occupied by type i under the threshold CR_i . The details are described as follows **Algorithm 2**. It is easy to see that the algorithm takes $O(|T||K||J|)$ time to complete the job assignment.

Algorithm 2 LPT-WB Heuristics

```

Let  $J^C$  be the unscheduled set of patients.  $I$  is the set of types.
 $N_i$  is the unassigned number of patients of type  $i$ . Set  $r_t = c_t, \forall t \in T$ .
for  $t$  in  $T$ , if  $r_t \geq 0$  do
     $A_i = 0, \forall i \in I$ 
    Sort beds into set  $K_t$  in increasing  $R_k$ .
    for  $k$  in  $K$  do
        for  $i$  in  $I$  do
            if all of the four conditions satisfied:
                1)  $N_i > 0$ , 2)  $r_t \geq s_j$ , 3)  $R_k + p_j \leq t$ , 4)  $\sum_{i' \leq i} A_{i'} < c_t CR_i$ 
                Set  $S_j = t, R_k = t + q_j + 1$ , and  $A_i = A_i + s_i$ 
                Remove  $j$  from  $J^C$ .  $N_i = N_i - 1$ 
            end for
        end for
    end for

```

4.3 LPT-TW heuristic

In this section, we propose a three-stage heuristic to deal with our problem. Since we mainly make two types of decisions, namely assignment, and schedule. Because of the complexity of the problem, these two decisions are involved, and no apparent structures are observed. A potential way of heuristic is to make sequential decisions step by step. Vairaktarakis and Cai (2003) proposed several heuristics to assign jobs to machines. We adopt the **Heuristic LPT** as our first step.

Heuristic LPT Vairaktarakis and Cai (2003)

- Step 1.** Let J^C be a set of patients sorted by longest $p_j + q_j$, following the LPT Rule.
- Step 2.** Pick the first job in J^C , and assign it to the first available bed.
- Step 3.** Go to **Step 2** until all patients are assigned to beds.

This heuristic assigns the patients with the LPT rule in order to balance the workload for each bed. For the second stage, we propose a First-Fit-based heuristic to schedule patients on each bed to occupy the operating room according to the given surgery block on a certain time window. We use $[t - a_t^k, t + b_t^k)$ to represent largest time periods available for bed k if the patient on bed k is scheduled for surgery on day t . If patient j leaves bed k at the end of period C_j , we have $R_k = C_j + 1$.

Heuristic First-Time Window

Initialize c_t corresponding to given surgery blocks. Set $r_t = c_t, \forall t = 1, \dots, T$.

Step 4. For each bed k , set $a_t^k = t$ and $b_t^k = T - t, \forall t = 1, \dots, T$.

Step 5. For j in J_k with type i , find the $t_j = \min\{t : r_t \geq s_j, a_t^k \geq p_j, b_t^k \geq q_j, \forall t = 1, \dots, T\}$.

Step 6. For $t = 1, \dots, T$, if $t < t_j$, set $b_t^k = \max(\min(t_j - p_j - t, b_t^k), 0)$. If $t > t_j$, set $a_t^k = \max(\min(t - t_j - q_j - 1, a_t^k), 0)$. If $t = t_j$, set $a_t^k = 0, b_t^k = 0$.

Step 7. Set $r_t = r_t - s_j, R_k = t_j + q_j + 1, S_j = t_j, C_j = t_j + q_j$.

Step 8. Go to **Step 5** until all patients in J_k is scheduled.

Step 9. Go to **Step 4** until patients in all beds are scheduled.

a_t^k and b_t^k are mainly used to characterize the forward and afterward time availability for bed k and surgery date t . In other words, $[t - a_t^k, t + b_t^k)$ is the time window available of bed k for surgery in day t . For example, if there exists t with $a_t^k = 1, b_t^k = 3$, then a patient who needs 2-day preoperative test or 5-day average recovery day are not able to receive surgery on day t if he/she is assigned to bed k . **Step 4** initializes status and assures that schedule of patients in bed k also follows *LPT* rule. **Step 5** finds the first surgery time window for patient j in bed k and settle the surgery date t_j for patient j . As a result, the time window $[t_j - p_j, t_j + q_j + 1)$ is no longer available for bed k . **Step 6** updates the rest of time windows of bed k for each t . **Step 7** updates the status of surgical resources r_t , release time of the bed k , surgery date and leaving date of patient j . The rest of steps schedule all the patients by iterating these steps.

Due to resource capacity and availability, there might be a situation where a patient j in bed k has to be delayed because of limited surgical resources or the long delay of the precious patient in the same bed. Therefore, although in stage 1, we assign patients to maintain a relatively balanced workload for each bed, some patients suffer from long delays. We adopt an adjustment for further improvement of the schedule by moving the severely delayed patients to other beds. We track the last surgery date for each bed and further define $S_{min} = \min_{k \in K} \{S^k : S^k \geq S_j, \forall j \in J_k\}$ and $S_{max} = \max\{S_j, \forall j \in J\}$ as the minimum and maximum last surgery date.

Heuristic Tail Adjustment

Step 10. Find the set of late patients $J^L = \{j : s_j > S_{min}, \text{ and } s_j \leq S_{max}\}$. Sort patients with LLT (Latest Leaving Time).

Step 11. For patient j in J^L , originally assigned to bed k_j^0 , and scheduled to t_j^0 find bed k_j and surgery date t_j , where $k_j, t_j = \min\{k, t : S_{min} \leq t \leq S_{max}, r_t \geq s_j, a_t^k \geq p_j, b_t^k \geq q_j, \forall t = 1, \dots, T, \forall k \in K\}$.

Step 12. If $t_j \geq t_j^0$, stop. Otherwise, remove j from $J_{k_j^0}$ and insert to J_{k_j} . Remove j from J^L . Go to **Step 6, 7, 10, 11** until that $t_j \geq t_j^0, \forall j \in J^L$.

This heuristic is designed to balance the workload of the tailing part of the schedule obtained from **Heuristic First-Time Window**, namely, the interval begins from when the first bed completes all surgery for patients from an original assignment from **Heuristic LPT**, till the last completion. In **Step 10**, for all the patients with surgery scheduled in this interval $(S_{min}, S_{max}]$, starting from the last patient in late set J^L , we try to reschedule his/her surgery date by moving to another bed. **Step 11** find the first available bed k_j and surgery date t_j for patient j . If the new operation day is better than the original, we move the patient j to the new bed in **Step 12**. The statuses are adjusted the same as in **Step 6, 7**. Repeat the steps till all potential improvements for late patients are processed. Note that if the surgery day of the last patient in J^L cannot be improved, adjustment for other patients would not affect the makespan and heuristic stops. Otherwise, we move on to reschedule immediate earlier patient.

In summary, the three-stage heuristic represents three types of decisions, namely assignment, scheduling, and rescheduling. They together constitute the **Algorithm 3**, called **LPT-TW Heuristic**, as a whole. It is easy to see that this heuristic takes $O(|K||J||T|)$ time to complete the sequencing process. Following the idea of balancing workloads of different types of patients in **Algorithm 2**, we propose **Algorithm 4**, called **LPT-TWT Heuristic**. This heuristic is adding a threshold criterion to **Algorithm 3** when assigning patients that if the surgical resource occupied by type i exceeds the threshold, we will not assign the patient this time to balance the workload of different types of patients. The calculation method of the threshold is the same as that in **Algorithm 2**. This heuristic also takes $O(|K||J||T|)$ time to complete the process because the threshold criterion does not increase the time complexity.

4.4 Composite index heuristic

A composite dispatching rule is an efficient tool in developing scheduling heuristics. The well-known ATC rule (Apparent Tardiness Cost Rule) was developed by Vepsalainen and Morton (1987) to deal with weighted tardiness cost by constructing an index combining WSPT (Weighted Shortes Processing Time) and LSR (Least Slack Remaining Processing Time) rule. Lee et al. (1997) propose the ATCS rule (Apparent Tardiness Cost with Setups) to deal with weighted tardiness with sequence-dependent setups. Su et al. (2017) further extended to the ATCF rule (Apparent Tardiness Cost with Flexibility) for applications with machine eligibility. Following a similar idea to combine different dispatching rules, we propose a Composite Index Heuristic for our problem. Note that different from weighted tardiness, we consider makespan as the objective. Therefore, the index is by nature significantly different from theirs. Specifically, I_{jt} is defined as a measure of the urgency of patient j at day t . It is a composite index to measure the critical level of unscheduled patients whenever a bed is released and available for the next patient. The job with a smaller index will be processed earlier. Specifically,

$$I_{jt} = \frac{ID_{jt'(j)} m_{t'(j)+p_j}}{p_j + q_j c_{t'(j)+p_j}} \tag{12}$$

For how this index represents the urgency of patients at a specific time, the two terms showcase three major features closely related to scheduling outcomes. The first term favors patients with longer processing time and less idle time. $p_j + q_j$ reflects the total in-bed duration of patient j . In-bed duration in the denominator shares the same idea of the LPT rule, indicating that patient who occupies the bed for a longer duration will have a smaller index value. $ID_{jt'(j)} = t'(j) - t$, where $t'(j) = \min\{t' : s_j \leq c_{t'+p_j}, t' \geq t\}$ and $c_{t'(j)+p_j}$ is the remaining surgical resource left in $t'(j) + p_j$ observed by provider at day t . The basic idea is to schedule each candidate patient to the earliest time with enough surgical resources. There might be bed idleness if the patient has to wait a while before admitted to the bed p_j days earlier than operation day. Moreover, less idleness leads to smaller I_{jt} value. In the second term, $c_{t'(j)+p_j}$ is the resources available at $t'(j) + p_j$ observed by provider at day t . $m_{t'(j)+p_j}$ is the number of beds that might be able to occupy the surgical resource at $t'(j) + p_j$ observed at time t . Specifically, when a patient j is scheduled to a future admission in day $t'(j)$ on bed k and a corresponding surgery day $t'(j) + p_j$, the remaining surgical resource at that time is reduced by s_j . Besides, the bed k is occupied from $t'(j)$ to $t'(j) + p_j + q_j$, during which for all time units with positive surgical resource, there will be one less bed potential to consume the remaining resource. The term $\frac{c_{t'(j)+p_j}}{m_{t'(j)+p_j}}$ measures the remaining resource that needs consumption per bed for surgery date $t'(j) + p_j$. The larger it is, the less likely the surgical resource is fully utilized at $t'(j) + p_j$. Then I_{jt} will be smaller. Therefore, the second term prioritizes the patient that leads to better utilization of surgical resources. In summary, this composite index combines the idea of Longest Processing Time, Minimum Idleness, and Largest Surgical Resource Per Bed Resource. Based on these intuitions, we propose the **Algorithm 5**. The details are as follows.

Algorithm 3 Composite Index Heuristic

Let J^c be the unscheduled set of jobs, and J_k be the set of jobs assigned to bed k .
 Initialize $R_k = 0, \forall k \in K, r_t = c_t$ and $m_t = |K|, \forall t \in T$
while $J^c \neq \emptyset$
 Set $k = \arg \min\{R_k, \forall k \in K\}, t = \min\{R_k, \forall k \in K\}$.
 for Each job j in J^c , **do**
 Calculate $ID_{jt'(j)}, I_{jt}$
 end for
 Choose the patient $j^* = \arg \min_j I_{jt}$. Remove it from J^c and add it to J_k .
 Update $r_{t'(j^*)+p_{j^*}} = r_{t'(j^*)+p_{j^*}} - s_{j^*}, m_{t'} = m_{t'} - 1, \forall t' \in [t, t'(j^*) + p_{j^*} + q_{j^*} + 1]$, and
 $R_k = t'(j^*) + p_{j^*} + q_{j^*} + 1$.
end while

We evaluate the measure I_{jt} for each unscheduled patient when a bed is released. A patient with the least I_{jt} value will be selected to occupy the bed successively. The status will be updated accordingly. Repeat the steps till all jobs are processed. It is easy to see that Algorithm 5 takes $O(|J|^2|T||K| \log |K|)$ time to complete the sequencing process.

4.5 MIP-based fix-and-dive improvement

Early computational results show that the performance of these five heuristics is not stable. The reason is that the coordinated schedule of the operating room and bed is a combination of $Pm|C_{max}$ and Bin-packing problem, which is a challenging problem. When the number of beds and that of surgical resources are balanced, the interaction is more involved, and the results of open-loop heuristics become less efficient. It is known that the mixed-integer problem in Sect. 2 is NP-hard, and preliminary results show that when many patients reach 60, the model cannot be solved to optimal within 7200s. In order to balance performance and computation time, we develop the MIP-based Fix-and-dive heuristics (**Algorithm 6-9**) to obtain further improvement. Steps are introduced as follows. First, the solutions of the heuristics mentioned above, **Algorithm 1-3 and 5**, are treated as initial solutions of **Algorithm 6-9** separately. We do not propose an improved heuristic based on **Algorithm 4** because its idea is from the same stream of **Algorithm 3** and early computational results show that the performance of **Algorithm 3** is better than that of **Algorithm 4**. Then, we add the variables z_{jk} to represent the assignment of patient j to bed k of initial solutions. Note that we fix the assignment of beds except for the ones with the longest completion time to improve the schedule of the beds with the worst performance. Finally, we solve the following mixed-integer programming (P2). Numerical results demonstrate that the MIP-based heuristics achieve near-optimal solutions with reasonable computation time.

$$\begin{aligned}
 \text{(P2)} \quad & \min C_{max} \\
 \text{s.t.} \quad & \sum_t x_{jk}^t \leq z_{jk}, & \forall j \in J, k \in K; \\
 & C_{max} \geq C_j, & \forall j \in J; \\
 & \sum_t \sum_k x_{jk}^t = 1, & \forall j \in J; \\
 & \sum_j x_{jk}^t \leq 1, & \forall k \in K, t \in T; \\
 & \sum_{i \neq j} \sum_{t'=t}^{t+p_j+q_j} x_{ik}^{t'} \leq M(1 - x_{jk}^t), & \forall j \in J, k \in K, t \in T; \\
 & \sum_k x_{jk}^t = y_j^{t+p_j}, & \forall j \in J, t \in T; \\
 & \sum_j y_j^t s_j \leq c_t, & \forall t \in T; \\
 & C_j \geq t y_j^t + q_j, & \forall j \in J, t \in T; \\
 & x_{jk}^t \in \{0, 1\}, & \forall j \in J, k \in K, t \in T; \\
 & y_j^t \in \{0, 1\}, & \forall j \in J, t \in T; \\
 & C_j \geq 0, & \forall j \in J.
 \end{aligned}$$

Algorithm 4 MIP-based Fix-and-divide Heuristics

Initialize: Let w_{jk} ($\forall k \in K, \forall j \in J$) be solutions of **Algorithm 1 (2, 3 and 5)** for **Algorithm 6-9** respectively. k^* is the index of bed with the longest completion time.

for $k \in K$ **do**

If $k = k^*$, $z_{jk} = 1, \forall j \in J$.

Otherwise, $z_{jk} = w_{jk}, \forall j \in J$.

end for

Solve the mixed-integer programming (P2) with z_{jk} .

5 Numerical results

In this section, we conduct a series of experiments to examine the model's performance and heuristics that we proposed. The mixed-integer program is coded by Python and is solved by Gurobi with a time limit of 7200s. The heuristics are coded by Python. All of the experiments are conducted using a PC with a 2.5GHz CPU.

The setting of our experiments is designed as follows. We assume three types of patients to represent patients with different diseases or severity. Patients of different types have a relatively steady population and different pre-surgery, surgery, and post-surgery time duration requirements. Because the more severe disease or the larger-scale operation often leads to longer surgical duration and length of the hospital stay. We set up three types of patients as follows. Type 1 patient with the most severe disease has the most extended time requirement with a 5-day pretest, 7-day recovery, and three-time units in surgery. Type 2 patient needs three days, five days, and two-time units before, after, and during surgery, respectively. Type 3 has a minor requirement with one day before and after surgery and one time unit during surgery. The proportion of the three types of patients is 1 : 3 : 6. Moreover, we fix the working hour of a physician or one medical team in a surgical block as ten-time units and assume one block a day is available. Multiple blocks a day can be equivalent to one block with longer working hours. As is often the case, a physician and his/her team will be assigned in several fixed blocks a week according to tactical level planning, which usually lasts weeks or even months. The patients accepted by different physicians will be scheduled separately. Our numerical study mainly examines the influence of the number of patients, the frequency of operation date, and the number of beds. The number of patients J is selected from [20, 40, 60, 80, 100], patient type is randomly generated according to the proportion. The weekly surgical frequency (i.e., the number of days the operating room is open per week) F varies from 1 to 5. The number of beds K belongs to [10, 15, 20].

5.1 Comparison of heuristics and MILP

This section compares the performance of MILP proposed in Sect. 3 and heuristics described in Sect. 4. We mainly focus on two kinds of performance measures. The first measure is the gap in makespan between heuristic solutions and optimal solutions solved by MILP. Specifically, $\text{Gap} = (C_{max}^h - C_{max}^*)/C_{max}^*$. The second performance

measure is the computation time (CPU) of different methods. The **A-Gap** means the average gap between heuristic results and MILP solutions, and **W-Gap** denotes the worst gap of the two solutions.

Table 6 displays the numerical results for small-size instances. All simple heuristics performs pretty well in terms of the average gap. Among the five heuristics, the LWBH and LTWTH have relatively large average gaps and worst gaps, which demonstrates the less steady performance. The performance of LRESH is in middle status. CIH performs slightly better than LTWH, which is the best.

Table 7 displays the computation time of MILP and simple heuristics. All five heuristics are fast to solve. Note that MILP can solve most small-size cases efficiently while it can be relatively time-consuming for certain instances even when patients reach 40. In view of the reasonable computation time and satisfactory performance of simple heuristics for small-size instances, we do not explore the MIP-based heuristics in the small tests.

For medium and large instances, we compare both simple heuristics and MIP-based heuristics with optimal solutions solved by MILP. From Table 8, CIH has the best overall performance for these five heuristics in that the average gap is within 10%, and it has the most negligible variance as parameter changes. LTWH and LTWTH have more significant average gaps than CIH but are also relatively stable in W-Gap. LRESH is quite unstable, and LWBH performs worst. They can obtain near-optimal solutions when surgery frequency is small and perform relatively poorly when frequency gets larger. Furthermore, there are common trends that all five heuristics share. When we fix other parameters, the performance improves as the number of patients increases, which perhaps results from the fact that the number of beds is relatively large compared to the number of patients. The more patients there are, the more risk pooling is involved. For another, the ratio of the number of beds and surgery frequency reflects the change of performance. In our experiment setting, 5 is a critical ratio. When the ratio is far from the critical ratio, one of the two resources is scarce, and the heuristics perform better. When the ratio is near critical, the performance is worse when the bed and surgical resources are relatively balanced.

From Tables 8, 9, and 10, the MIP-based heuristics perform much better than the open-loop heuristics at the cost of longer computation time, described in Fig. 3. Almost all of the performance gaps are under 10%. Computation times are significantly small compared with MILP and grows as number of patients increases. Although we fix the assignment for most beds, as the number of patients increases, the quantities of free assignments still increase.

To sum up, the heuristics we proposed are practical. For hospitals, simple heuristics can be used to get a good solution quickly for small-scale problems. When the problem is large-scale or simple heuristics effect is not good, the manager can use the improved algorithm to get a good scheme reasonably.

5.2 Parameter sensitivity

This section analyzes the influence of parameters on the objective, specifically the frequency of surgery and the number of beds.

Table 6 Performance gaps of simple heuristics for small-sized instances

Instance	K	F	LRESH		LWBH		LWTH		LWTH		CIH	
			A-Gap	W-Gap	A-Gap	W-Gap	A-Gap	W-Gap	A-Gap	W-Gap	A-Gap	W-Gap
20	10	2	0.040	0.200	0.100	0.200	0.040	0.200	0.266	0.467	0.060	0.400
20	15	2	0.000	0.000	0.080	0.200	0.000	0.000	0.100	0.200	0.000	0.000
20	20	2	0.000	0.000	0.080	0.200	0.080	0.200	0.080	0.200	0.000	0.000
20	10	3	0.103	0.143	0.176	0.400	0.007	0.071	0.146	0.313	0.054	0.143
20	15	3	0.141	0.143	0.141	0.143	0.000	0.000	0.100	0.286	0.043	0.143
20	10	4	0.078	0.133	0.135	0.500	0.000	0.000	0.134	0.214	0.099	0.286
20	15	4	0.078	0.133	0.078	0.133	0.000	0.000	0.057	0.214	0.029	0.071
20	20	3	0.000	0.000	0.057	0.143	0.057	0.143	0.086	0.143	0.000	0.000
20	20	4	0.000	0.000	0.000	0.000	0.000	0.000	0.043	0.071	0.000	0.000
40	10	2	0.175	0.316	0.191	0.440	0.132	0.280	0.174	0.440	0.138	0.280
40	15	2	0.000	0.000	0.156	0.316	0.040	0.105	0.171	0.316	0.058	0.211
40	20	2	0.000	0.000	0.156	0.316	0.184	0.316	0.166	0.316	0.000	0.000
40	10	3	0.136	0.455	0.345	0.429	0.083	0.182	0.112	0.182	0.122	0.208
40	15	3	0.168	0.563	0.417	0.563	0.092	0.188	0.249	0.353	0.109	0.263
40	20	3	0.000	0.000	0.211	0.313	0.249	0.400	0.297	0.400	0.000	0.000
40	10	4	0.143	0.292	0.204	0.409	0.070	0.158	0.107	0.263	0.125	0.263
40	15	4	0.429	0.500	0.423	0.500	0.080	0.200	0.230	0.412	0.205	0.333
40	20	4	0.389	0.500	0.416	0.500	0.141	0.214	0.198	0.294	0.006	0.059
			0.104		0.187		0.070		0.151		0.058	

Table 7 CPU time of MILP and simple heuristics for small-sized instances

Instance			MILP	LRESH	LWBH	LTWH	LTWTH	CIH
J	K	F						
20	10	2	20	0.01	0.02	0.02	0.01	0.01
20	15	2	31	0.01	0.03	0.02	0.02	0.02
20	20	2	42	0.01	0.04	0.02	0.01	0.01
20	10	3	25	0.01	0.03	0.02	0.01	0.02
20	15	3	37	0.01	0.04	0.03	0.02	0.02
20	10	4	29	0.01	0.06	0.03	0.02	0.02
20	15	4	43	0.01	0.09	0.04	0.03	0.03
20	20	3	52	0.01	0.05	0.04	0.01	0.02
20	20	4	38	0.01	0.10	0.05	0.02	0.02
40	10	2	78	0.03	0.03	0.02	0.01	0.02
40	15	2	115	0.03	0.04	0.03	0.01	0.02
40	20	2	149	0.03	0.05	0.03	0.01	0.02
40	10	3	342	0.03	0.04	0.03	0.02	0.02
40	15	3	144	0.03	0.05	0.03	0.03	0.03
40	20	3	188	0.03	0.06	0.04	0.02	0.04
40	10	4	4582	0.03	0.07	0.03	0.03	0.03
40	15	4	176	0.03	0.09	0.04	0.03	0.04
40	20	4	235	0.03	0.11	0.05	0.03	0.05

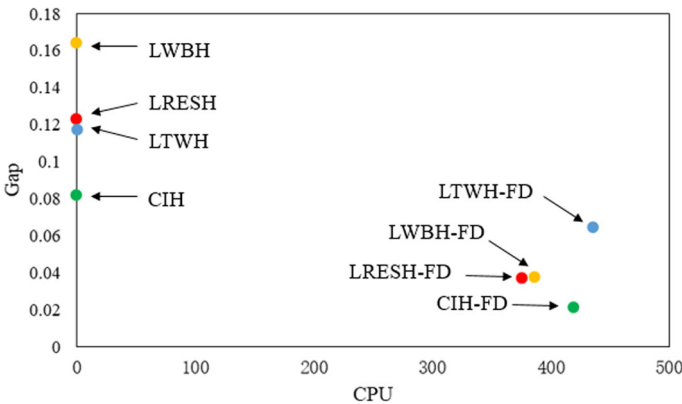


Fig. 3 Statistical results of compared constructive heuristics for medium-and-large-sized instances with A-GAP and CPU

Figure 4 displays the impact of the number of beds upon Cmax with other parameters fixed. Figure 5 indicates the impact of the surgery frequency upon Cmax with other parameters fixed. Note that we do not cover all the parameters in our numerical study because too few or too many resources compared to the other is trivial, as explained in Sect. 3. For the sensitivity analysis, however, when we expand the range of

Table 8 Performance gaps of simple heuristics for medium-and-large-sized instances

Instance			LRESH		LWBH		LTWH		LTWTH		CIH	
J	K	F	A-Gap	W-Gap	A-Gap	W-Gap	A-Gap	W-Gap	A-Gap	W-Gap	A-Gap	W-Gap
60	10	2	0.191	0.243	0.138	0.219	0.135	0.219	0.136	0.243	0.155	0.219
60	15	2	0.000	0.000	0.075	0.182	0.055	0.167	0.079	0.182	0.016	0.100
60	20	2	0.000	0.000	0.075	0.182	0.074	0.167	0.124	0.200	0.016	0.100
60	10	3	0.213	0.394	0.268	0.333	0.131	0.188	0.131	0.188	0.110	0.152
60	15	3	0.145	0.455	0.383	0.458	0.190	0.292	0.236	0.409	0.172	0.208
60	20	3	0.018	0.091	0.116	0.250	0.223	0.273	0.236	0.273	0.036	0.091
60	10	4	0.105	0.226	0.181	0.310	0.076	0.125	0.076	0.125	0.111	0.188
60	15	4	0.181	0.273	0.257	0.591	0.120	0.250	0.131	0.250	0.163	0.333
60	20	4	0.444	0.563	0.487	0.563	0.272	0.333	0.370	0.412	0.106	0.167
80	10	2	0.153	0.205	0.109	0.152	0.112	0.150	0.129	0.213	0.126	0.167
80	15	2	0.000	0.000	0.031	0.075	0.054	0.100	0.059	0.150	0.000	0.000
80	20	2	0.000	0.000	0.031	0.075	0.069	0.150	0.078	0.150	0.010	0.100
80	10	3	0.198	0.262	0.169	0.225	0.122	0.209	0.113	0.156	0.098	0.175
80	15	3	0.195	0.355	0.328	0.400	0.159	0.241	0.201	0.258	0.155	0.200
80	20	3	0.012	0.115	0.049	0.129	0.164	0.231	0.187	0.231	0.039	0.115
80	10	4	0.046	0.079	0.131	0.238	0.046	0.154	0.059	0.154	0.056	0.154
80	15	4	0.139	0.267	0.205	0.357	0.094	0.192	0.093	0.172	0.109	0.192
80	20	4	0.334	0.455	0.349	0.455	0.161	0.318	0.224	0.318	0.095	0.136
100	10	2	0.085	0.125	0.085	0.125	0.120	0.224	0.110	0.224	0.115	0.176
100	15	2	0.000	0.000	0.004	0.039	0.024	0.043	0.033	0.056	0.023	0.085
100	20	2	0.000	0.000	0.004	0.039	0.063	0.128	0.076	0.128	0.029	0.085
100	10	3	0.176	0.260	0.137	0.175	0.139	0.180	0.140	0.180	0.093	0.135
100	15	3	0.208	0.361	0.273	0.361	0.164	0.250	0.152	0.216	0.141	0.194
100	20	3	0.018	0.091	0.029	0.091	0.115	0.182	0.162	0.273	0.035	0.091
100	10	4	0.036	0.073	0.052	0.093	0.030	0.056	0.031	0.054	0.023	0.060
100	15	4	0.103	0.182	0.165	0.250	0.087	0.154	0.074	0.152	0.093	0.182
100	20	4	0.321	0.385	0.307	0.385	0.172	0.250	0.188	0.267	0.089	0.154
			0.123		0.164		0.117		0.134		0.082	

parameters, we can figure out the trend, take Fig. 4 as an example, that when surgery frequency is fixed, as the number of beds increases, Cmax will decrease rapidly at first, then slow down and finally stabilize at a certain level. Because when the beds are much more abundant than the surgical resource, the latter resource becomes a bottleneck and finally determines the whole system’s performance. On this condition, an increase in the surgery frequency would improve the utilization of beds and further lower the Cmax. Figure 5 shows similar results. This also provides some insights into medical management. The hospital should determine appropriate resources required by considering coordination utilization of hospital beds and operating rooms. If the

Table 9 Performance gaps of MIP-based heuristics for medium-and-large-sized instances

Instance			LRESH-FD		LWB-FD		LTW-FD		CIH-FD	
J	K	F	A-Gap	W-Gap	A-Gap	W-Gap	A-Gap	W-Gap	A-Gap	W-Gap
60	10	2	0.096	0.156	0.032	0.067	0.093	0.182	0.027	0.067
60	15	2	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
60	20	2	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
60	10	3	0.035	0.111	0.035	0.111	0.035	0.111	0.050	0.111
60	15	3	0.051	0.130	0.055	0.130	0.150	0.208	0.052	0.091
60	20	3	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
60	10	4	0.046	0.094	0.022	0.094	0.081	0.207	0.019	0.125
60	15	4	0.070	0.136	0.041	0.120	0.074	0.222	0.062	0.167
60	20	4	0.009	0.063	0.068	0.167	0.034	0.222	0.011	0.056
80	10	2	0.127	0.159	0.059	0.109	0.070	0.150	0.030	0.050
80	15	2	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
80	20	2	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
80	10	3	0.042	0.093	0.062	0.111	0.106	0.200	0.028	0.093
80	15	3	0.069	0.161	0.067	0.138	0.125	0.161	0.093	0.161
80	20	3	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
80	10	4	0.040	0.105	0.061	0.105	0.086	0.184	0.019	0.071
80	15	4	0.055	0.192	0.084	0.219	0.129	0.250	0.046	0.107
80	20	4	0.004	0.045	0.080	0.167	0.111	0.208	0.017	0.087
100	10	2	0.134	0.176	0.058	0.121	0.078	0.176	0.013	0.039
100	15	2	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
100	20	2	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
100	10	3	0.045	0.080	0.049	0.080	0.122	0.185	0.033	0.080
100	15	3	0.067	0.132	0.032	0.111	0.091	0.132	0.055	0.111
100	20	3	0.000	0.000	0.000	0.000	0.022	0.111	0.000	0.000
100	10	4	0.014	0.054	0.042	0.125	0.066	0.125	0.003	0.060
100	15	4	0.114	0.167	0.084	0.167	0.096	0.216	0.037	0.083
100	20	4	0.000	0.077	0.087	0.241	0.131	0.310	0.000	0.067
			0.038		0.038		0.063		0.022	

manager only focuses on one resource, it will result in redundancy and waste of another resource.

Another interesting result is that the point that slope begins to slow down in coincidence with the ratio of beds and frequency, as we mentioned in numerical results. This indicates that when the two resources are relatively balanced, we need to pay more attention to the coordinated utilization of beds and operating rooms, highlighting our study’s importance.

Table 10 CPU time of MILP and all of heuristics for medium-and-large-sized instances

Instance	MILP			LRESH	LWB	LTW	LTWP	CIH	LRESH FD	LWB FD	LTW FD	CIH FD
	J	K	F									
60	10	2	242	0.03	0.04	0.02	0.02	0.07	97	104	167	125
60	15	2	198	0.04	0.05	0.02	0.02	0.07	223	235	149	180
60	20	2	269	0.04	0.05	0.03	0.02	0.07	296	311	196	245
60	10	3	2794	0.04	0.05	0.03	0.03	0.07	98	107	164	129
60	15	3	1389	0.04	0.07	0.03	0.04	0.07	142	238	236	184
60	20	3	338	0.05	0.07	0.04	0.05	0.07	297	312	196	247
60	10	4	7200	0.04	0.08	0.03	0.04	0.07	103	205	206	143
60	15	4	5891	0.05	0.07	0.05	0.06	0.07	147	255	245	196
60	20	4	3677	0.06	0.12	0.05	0.07	0.07	300	319	209	254
80	10	2	2957	0.05	0.05	0.02	0.03	0.12	190	304	374	222
80	15	2	378	0.05	0.06	0.03	0.03	0.12	397	417	402	324
80	20	2	499	0.06	0.07	0.04	0.04	0.13	554	336	541	660
80	10	3	5477	0.05	0.06	0.03	0.04	0.11	192	290	172	221
80	15	3	2671	0.06	0.07	0.05	0.05	0.12	255	261	410	327
80	20	3	633	0.07	0.08	0.03	0.05	0.12	557	556	536	698
80	10	4	7200	0.06	0.10	0.04	0.05	0.11	202	316	307	253
80	15	4	7200	0.07	0.08	0.05	0.06	0.11	265	267	414	336
80	20	4	5720	0.08	0.14	0.07	0.09	0.12	578	537	557	723
100	10	2	3137	0.07	0.07	0.03	0.03	0.18	273	432	471	516
100	15	2	1376	0.07	0.08	0.04	0.04	0.19	635	406	624	744
100	20	2	2031	0.08	0.09	0.05	0.06	0.22	560	569	820	1097
100	10	3	7200	0.07	0.09	0.04	0.05	0.18	438	431	434	537
100	15	3	4780	0.08	0.12	0.05	0.06	0.18	655	654	621	759
100	20	3	1626	0.09	0.10	0.04	0.06	0.19	828	836	822	989
100	10	4	7200	0.07	0.11	0.05	0.06	0.18	323	508	563	593
100	15	4	7200	0.08	0.13	0.06	0.07	0.17	652	688	642	819
100	20	4	7200	0.10	0.15	0.09	0.09	0.17	882	548	846	1013

5.3 Managerial insight

Traditional Operating Room Scheduling has historically been centered on maximizing the utilization of the operating room, a valuable and costly resource within the hospital infrastructure. However, we contend that the significance of hospital beds, often overlooked in this traditional paradigm, should be considered. Hospital beds represent a dual-faceted, scarce resource characterized by prolonged occupancy and serving as a prerequisite for multiple surgical procedures. The timely admission of patients and the effective utilization of hospital beds play a pivotal role in the success of surgical interventions.

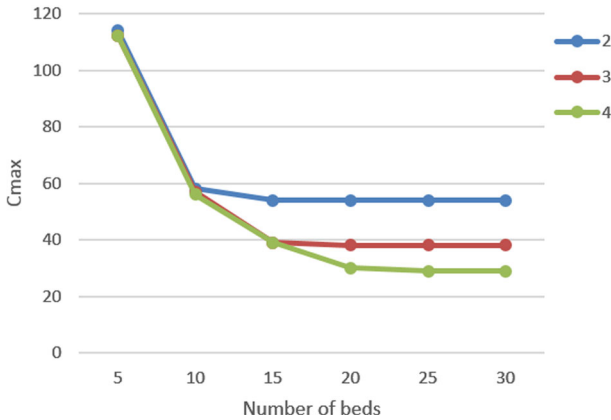


Fig. 4 Cmax with different number of beds

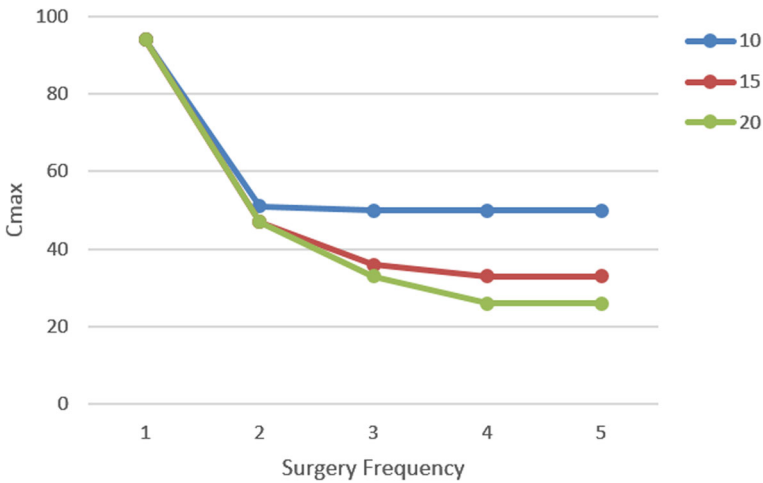


Fig. 5 Cmax with different surgery frequency

When formulating an optimal resource allocation strategy, hospital managers must consider the coordinated utilization of hospital beds and operating rooms. A disproportionate focus on one resource over the other may result in redundancy and wastage, potentially leading to an imbalance in resource allocation. Such an imbalance could deplete operating room resources while underutilizing hospital bed resources, compromising the healthcare system’s overall operational efficiency. Hence, we emphasize adopting a comprehensive and coordinated approach to resource utilization, ensuring the optimal use of operating rooms and hospital beds.

Furthermore, in scenarios with relative equilibrium between hospital bed and operating room utilization, our research argues for heightened attention to the coordinated utilization of these resources. This perspective underscores the critical importance of our study, particularly in high-ranking hospitals in first-tier cities. These hospitals often have inadequate medical resources due to prolonged patient waiting lists. In

such a context, optimizing the utilization of hospital beds and operating rooms not only enhances overall efficiency but also more effectively meets the needs of patients. Implementing a coordinated resource utilization strategy becomes particularly urgent in alleviating the strain on healthcare services in environments with scarce medical resources, thereby upholding the stability and quality of healthcare services.

6 Conclusion and future studies

In this paper, we explore the joint surgery scheduling and bed assignment to improve the utilization of both resources. We show that the joint decision is of practical importance. A mixed-integer model is developed to minimize the maximum completion time. We apply the classic LPT rule, workload balance, composite index and so on to propose heuristics (LRESH, LWBH, LTWH, LTWTH and CIH) dealing with the coordinated scheduling of hospital beds and operating rooms. They perform well in small-size instances and are unstable for large-size instances. Furthermore, we propose improved algorithms, called LRESH-FD, LWBH-FD, LTWH-FD, and CIH-FD. We structured a way to simplify the problem and got near-optimal solutions even for large-size instances.

The contributions of this work are two-fold. First, patient scheduling is a vital planning activity that hospital managers must consider. Besides, resources, including operating rooms and hospital beds, are critical in health care. In this sense, we jointly study the surgery scheduling and bed assignment problem. Second, solutions that efficiently leads to near optimal results can benefit the planning department. From this perspective, we develop practical heuristics, and computational experiments show that our proposed algorithms perform well in computational time and solution quality.

Our work can be extended from several aspects. First, the optimization model and the solution approach developed in this study may be the basis for further developments in a more effective and efficient surgery system by considering different objectives and constraints. Second, no-shows, service delays, and resource availability are widely witnessed in practice. Thus, it is meaningful to consider the problem with a dynamic version in real-world situations.

Funding Zhaohui Li is supported in part by the China Postdoctoral Science Foundation GZC20230378 and the National Natural Science Foundation of China 72301055. Haiyue Yu is supported in part by National Health Commission Hospital Management Institute Healthcare Quality (Evidence-Based) Management Research Project YLZLXZ23H004, Shanghai Shengkang Hospital Development Center Healthcare Management Project SHDC12023628, and Traditional Chinese Medicine Science and Technology Development Project of Shanghai Medical Innovation & Development Foundation WL-HBQN-2022019K.

Data availability The datasets generated during and/or analysed during the current study are available from the corresponding author on reasonable request.

Declarations

Conflict of interest No potential Conflict of interest was reported by the author(s).

References

- Adan IJBF, Vissers JMH (2002) Patient mix optimisation in hospital admission planning: a case study. *Int J Op Oroduction Manag* 22(4):445–461
- Barz C, Rajaram K (2015) Elective patient admission and scheduling under multiple resource constraints. *Prod Oper Manag* 24(12):1907–1930
- Bastos LSL, Marchesi JF, Hamacher S, Fleck JL (2019) A mixed integer programming approach to the patient admission scheduling problem. *Eur J Op Res* 273(3):831–840
- Behmanesh R, Zandieh M (2019) Surgical case scheduling problem with fuzzy surgery time: an advanced bi-objective ant system approach. *Knowl-Based Syst* 186:104913
- Beliën J, Demeulemeester E (2007) Building cyclic master surgery schedules with leveled resulting bed occupancy. *Eur J Oper Res* 176(2):1185–1204
- Belkhamza M, Jarbouï B, Masmoudi M (2018) Two metaheuristics for solving no-wait operating room surgery scheduling problem under various resource constraints. *Comput Ind Eng* 126:494–506
- Blake JT, Carter MW (2002) A goal programming approach to strategic resource allocation in acute care hospitals. *Eur J Op Res* 140(3):541–561
- Braglia M, Petroni A (1999) Data envelopment analysis for dispatching rule selection. *Prod Plan Control* 10(5):454–461
- Cappanera P, Visintin F, Banditori C (2014) Comparing resource balancing criteria in master surgical scheduling: a combined optimisation-simulation approach. *Int J Prod Econ* 158:179–196
- Castro PM, Marques I (2015) Operating room scheduling with generalized disjunctive programming. *Comput Op Res* 64:262–273
- Ceschia S, Schaerf A (2011) Local search and lower bounds for the patient admission scheduling problem. *Comput Op Res* 38(10):1452–1463
- Ceschia S, Schaerf A (2016) Dynamic patient admission scheduling with operating room constraints, flexible horizons, and patient delays. *J Sched* 19(4):377–389
- Dehua X, Sun K, Li H (2008) Parallel machine scheduling with almost periodic maintenance and non-preemptive jobs to minimize makespan. *Comput Oper Res* 35(4):1344–1349
- Dellaert N, Jeunet J (2017) A variable neighborhood search algorithm for the surgery tactical planning problem. *Comput Oper Res* 84:216–225
- Demeester P, Souffriau W, De Causmaecker P, Berghe GV (2010) A hybrid tabu search algorithm for automatically assigning patients to beds. *Artif Intell Med* 48(1):61–70
- Denton B, Viapiano J, Vogl A (2007) Optimization of surgery sequencing and scheduling decisions under uncertainty. *Health Care Manag Sci* 10(1):13–24
- Erdogan SA, Denton BT, Cochran JJ, Cox LA, Keskinocak P, Kharoufeh J, Smith J (2011) Surgery planning and scheduling. In: *Wiley Encyclopedia of Operations Research and Management Science*. Wiley, Hoboken, NJ, 2011
- Erdogan SA, Gose A, Denton BT (2015) Online appointment sequencing and scheduling. *IIE Trans* 47(11):1267–1286
- Fei H, Meskens N, Chu C (2010) A planning and scheduling problem for an operating theatre using an open scheduling strategy. *Comput Ind Eng* 58(2):221–230
- Guerrero F, Guido R (2011) Operational research in the management of the operating theatre: a survey. *Health Care Manag Sci* 14(1):89–114
- Guido R, Conforti D (2017) A hybrid genetic approach for solving an integrated multi-objective operating room planning and scheduling problem. *Comput Oper Res* 87:270–282
- Hamid M, Nasiri MM, Werner F, Sheikahmadi F, Zhalechian M (2019) Operating room scheduling by considering the decision-making styles of surgical team members: a comprehensive approach. *Comput Oper Res* 108:166–181
- Harper PR (2002) A framework for operational modelling of hospital resources. *Health Care Manag Sci* 5(3):165–173
- Hof S, Fügener A, Schoenfelder J, Brunner JO (2017) Case mix planning in hospitals: a review and future agenda. *Health Care Manag Sci* 20(2):207–220
- Holm LB, Lurås H, Dahl FA (2013) Improving hospital bed utilisation through simulation and optimisation: with application to a 40% increase in patient volume in a norwegian general hospital. *Int J Med Inform* 82(2):80–89
- Hoon Lee Y, Bhaskaran K, Pinedo M (1997) A heuristic to minimize the total weighted tardiness with sequence-dependent setups. *IIE Trans* 29(1):45–52

- Hosseini N, Taaffe KM (2015) Allocating operating room block time using historical caseload variability. *Health Care Manag Sci* 18(4):419–430
- Hughes WL, Soliman SY (1985) Short-term case mix management with linear programming. *Hospital Health Serv Adm* 30(1):52–60
- Huiqiao S, Pinedo M, Wan G (2017) Parallel machine scheduling with eligibility constraints: a composite dispatching rule to minimize total weighted tardiness. *Naval Res Logist (NRL)* 64(3):249–267
- Hulshof PJH, Kortbeek N, Boucherie RJ, Hans EW, Bakker PJM (2012) Taxonomic classification of planning decisions in health care: a structured review of the state of the art in *or/ms*. *Health Systems* 1(2):129–175
- Jebali A, Hadj Alouane AB, Ladet P (2006) Operating rooms scheduling. *Int J Prod Econ* 99(1–2):52–62
- Lane DC, Monefeldt C, Rosenhead JV (2000) Looking in the wrong place for healthcare improvements: a system dynamics study of an accident and emergency department. *J Oper Res Soc* 51:518–531
- Macario A, Vitez TS, Dunn B, McDonald T (1995) Where are the costs in perioperative care?: Analysis of hospital costs and charges for inpatient surgical care. *J Am Soc Anesthesiol* 83(6):1138–1144
- Mahdi Nasiri M, Rahvar M (2017) A two-step multi-objective mathematical model for nurse scheduling problem considering nurse preferences and consecutive shifts. *Int J Serv Oper Manag* 27(1):83–101
- Mannino C, Nilssen EJ, Nordlander TE (2012) A pattern based, robust approach to cyclic master surgery scheduling. *J Sched* 15(5):553–563
- Marchesi JF, Pacheco MAC (2016) A genetic algorithm approach for the master surgical schedule problem. In: 2016 IEEE Conference on Evolving and Adaptive Intelligent Systems (EAIS), pp 17–21. IEEE, 2016
- Marques I, Captivo ME, Pato MV (2014) Scheduling elective surgeries in a portuguese hospital using a genetic heuristic. *Oper Res Health Care* 3(2):59–72
- Michael LP (2018) *Scheduling: theory, algorithms, and systems*. Springer, Berlin
- Moosavi A, Ebrahimnejad S (2020) Robust operating room planning considering upstream and downstream units: a new two-stage heuristic algorithm. *Comput Ind Eng* 143:106387
- Ni C, Zhi-hui Z, Jun Z, Ou L, Hai-lin L (2010) A genetic algorithm for the optimization of admission scheduling strategy in hospitals. In: IEEE congress on evolutionary computation, pp 1–5. IEEE
- Penn ML, Potts CN, Harper PR (2017) Multiple criteria mixed-integer programming for incorporating multiple factors into the development of master operating theatre timetables. *Eur J Oper Res* 262(1):194–206
- Pham D-N, Klinkert A (2008) Surgical case scheduling as a generalized job shop scheduling problem. *Eur J Oper Res* 185(3):1011–1025
- Roshanaei V, Booth KEC, Aleman DM, Urbach DR, Beck JC (2020) Branch-and-check methods for multi-level operating room planning and scheduling. *Int J Prod Econ* 220:107433
- Salma M, Said K, Abderrahman A, Ahmed EH, Alaoui (2022) A two-stage robust optimization approach for the master surgical schedule problem under uncertainty considering downstream resources. *Health Care Manag Sci* 25(1):63–88
- Santos D, Marques I (2022) Designing master surgery schedules with downstream unit integration via stochastic programming. *Eur J Oper Res* 299(3):834–852
- Saremi A, Jula P, Elmekawy T, Wang GG (2013) Appointment scheduling of outpatient surgical services in a multistage operating room department. *Int J Prod Econ* 141(2):646–658
- Testi A, Tanfani E (2009) Tactical and operational decisions for operating room planning: efficiency and welfare implications. *Health Care Manag Sci* 12(4):363–373
- Testi A, Tanfani E, Torre G (2007) A three-phase approach for operating theatre schedules. *Health Care Manag Sci* 10(2):163–172
- Vairaktarakis GL, Cai X (2003) The value of processing flexibility in multipurpose machines. *IIE Trans* 35(8):763–774
- Vanberkel PT, Boucherie RJ, Hans EW, Hurink JL, Van Lent WAM, Van Harten WH (2011) An exact approach for relating recovering surgical patient workload to the master surgical schedule. *J Oper Res Soc* 62(10):1851–1860
- Vancroonenburg W, De Causmaecker P, Vanden Berghe G (2012) Patient-to-room assignment planning in a dynamic context. In: Proceedings of the 9th international conference on the practice and theory of automated timetabling (PATAT-2012), pp 193–208. Citeseer
- Vepsalainen APJ, Morton TE (1987) Priority rules for job shops with weighted tardiness costs. *Manag Sci* 33(8):1035–1047

- Visser JMH, Adan IJBF, Bekkers JA (2005) Patient mix optimization in tactical cardiothoracic surgery planning: a case study. *IMA J Manag Math* 16(3):281–304
- Visser JMH, Adan IJBF, Dellaert NP (2007) Developing a platform for comparison of hospital admission systems: an illustration. *Eur J Oper Res* 180(3):1290–1301
- Wang S, Huiqiao S, Wan G (2015) Resource-constrained machine scheduling with machine eligibility restriction and its applications to surgical operations scheduling. *J Comb Optim* 30(4):982–995
- Wayne Meredith J, High KP, Freischlag JA (2020) Preserving elective surgeries in the covid-19 pandemic and the future. *Jama* 324(17):1725–1726
- Xiang W, Yin J, Lim G (2015) An ant colony optimization approach for solving an operating room surgery scheduling problem. *Comput Ind Eng* 85:335–345
- Yu H, Li J, Chen X, Niu W, Sang H (2022) An improved multi-objective imperialist competitive algorithm for surgical case scheduling problem with switching and preparation times. *Clust Comput* 25:1–26
- Zhong L, Luo S, Lidong W, Lin X, Yang J, Tang G (2014) A two-stage approach for surgery scheduling. *J Comb Optim* 27(3):545–556
- Zhu T, Luo L, Shen W, Xu X, Kou R (2020) Admission scheduling of inpatients by considering two inter-related resources: beds and operating rooms. *Optimization* 71:1–39

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

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.