Health Care Manag Sci (2018) 21:376–392
DOI 10.1007/s10729-016-9390-2

Two-MILP models for scheduling elective surgeries within a private healthcare facility

Hejer Khlif Hachicha[1] · Farah Zeghal Mansour[1]

Received: 18 May 2016 / Accepted: 20 October 2016 / Published online: 5 November 2016

Abstract This paper deals with an Integrated Elective Surgery-Scheduling Problem (IESSP) that arises in a privately operated healthcare facility. It aims to optimize the resource utilization of the entire surgery process including pre-operative, per-operative and post-operative activities. Moreover, it addresses a specific feature of private facilities where surgeons are independent service providers and may conduct their surgeries in different private healthcare facilities. Thus, the problem requires the assignment of surgery patients to hospital beds, operating rooms and recovery beds as well as their sequencing over a 1-day period while taking into account surgeons' availability constraints. We present two Mixed Integer Linear Programs (MILP) that model the IESSP as a three-stage hybrid flow-shop scheduling problem with recirculation, resource synchronization, dedicated machines, and blocking constraints. To assess the empirical performance of the proposed models, we conducted experiments on real-world data of a Tunisian private clinic: *Clinique Ennasr* and on randomly generated instances. Two criteria were minimised: the patients' average length of stay and the number of patients' overnight stays. The computational results show that the proposed models can solve instances with up to 44 surgical cases in a reasonable CPU time using a general-purpose MILP solver.

✉ Hejer Khlif Hachicha
khlif.hejer@yahoo.fr

Farah Zeghal Mansour
farah.zeghal@enit.rnu.tn

[1] UR-OASIS, Ecole Nationale d'Ingénieurs de Tunis, Université de Tunis El Manar, 1002 Tunis, Tunisia

Keywords Private healthcare facility · Surgery process · Operating theatre · Hybrid flow shop scheduling · Recirculation · Resource synchronization · Dedicated machines · Mixed integer linear program

1 Introduction

The private healthcare has been in a continual growth over recent years. Indeed, more people are now using private healthcare facilities rather than public, though the costs in the latter are much more affordable. Moreover, medical tourism is thriving, and patients are travelling from developed countries (largely from the United States, Canada, and Western Europe) to less developed ones mainly in Asia and Latin America for reduced-wait time and/or high-quality medical care at affordable prices [1].

To meet this growing demand, many private healthcare facilities have emerged and global competition is strengthened. In this context, private healthcare facilities increasingly strive for optimising their resource utilisation and thus improving their care services quality while increasing their profits. For that aim, they usually focus on the optimization of the most cost-intensive and productive unit: the operating theatre.

In practice, there are two types of surgery operations: *elective operations* that are planned in advance, and *non-elective* or *emergency operations* that arrive unexpectedly and need to be performed urgently. In both cases, patients can be hospitalized for either less than 24 h (called outpatients) or several days (called inpatients).

Unlike the public sector where surgeons work for a specific hospital, in the private sector, surgeons are independent service providers and may refer their patients to several private healthcare facilities. Thus, to increase the clinic workload, surgeon's dependent processes have to be facilitated (e.g., surgery scheduling processes) [2].

In this paper, we investigate the Integrated Elective Surgery-Scheduling Problem (IESSP) that arises in a privately operated healthcare facility. Moreover, we consider the entire surgery process that comprises the following three phases [3, 4]:

- *The pre-operative phase*: It commonly includes patients' admission and allows performing tests, preoperative fasting, patient preparation, etc.
- *The per-operative phase*: It begins when the patient is transferred to the operating theatre and covers her/his stay there until her/his transfer to the corresponding hospital bed. During this phase, surgery operation and recovery care are performed.
- *The post-operative phase*: It refers to the management of the patient cares after a surgery. It starts when the patient is transferred to her/his hospital bed and lasts until she/he leaves it.

The contribution of this paper is twofold:

1. *The integration of the hospital beds management within the daily operating theatre scheduling problem*: Most previous research focus on the operating theatre scheduling in the context of public healthcare hospitals. They consider diverse human and material resource constraints related to the operating theatre but do not consider hospital ward capacity. Consequently, the generated schedules may be infeasible when the beds are scarce resources as patients may not be admitted for bed shortage. Thus, the hospital ward should be managed efficiently together with the operating room scheduling to increase the number of accepted surgical cases and reduce the number of refused admissions. In the private context, managing the hospital beds and the operating theatre simultaneously is becoming compulsory since it may increase their profits and improve the quality of their services for both patients and surgeons.
2. *The development of effective and flexible MILP models to solve exactly the IESSP*: The scheduling problem is known to be NP-hard and is commonly solved using approximate methods or sophisticated exact approaches. In this paper, we develop two *effective* mixed integer linear models that can be efficiently solved to optimality using a commercial software package without requiring the implementation of tailored and sophisticated algorithms. The first model is a network-flow MILP that uses the same variable to express the two problem decisions: patient sequencing and resource assignment. The second model is an assignment based MILP that defines two different variables that express these two decisions separately. Moreover, the proposed models are *flexible* and can be readily modified to accommodate several extensions. Thus, they may open the door for the development of new efficient solution methodologies to model uncertainty and robustness features in healthcare scheduling problems.

This paper is organized as follows: Section 2 presents a review of the literature pertaining to the surgery-scheduling problem. Section 3 describes the considered integrated elective surgery-scheduling problem. Section 4 introduces two mixed integer linear models and provides some extensions. In Section 5, computational experiments are presented and discussed. Finally, Section 6 draws conclusions and gives some perspective topics.

2 Literature review

Patient and healthcare-resource planning and scheduling problems have been extensively studied and referenced in traditional Operational Research journals during these last decades. In particular, operating theatre has deserved special attention since it is considered to be the main engine of the hospital.

Two major optimization problems are addressed: (1) operating room planning (ORP), also called ***advanced scheduling*** that consists on assigning a surgery date to patients, and (2) operating room scheduling (ORS), also called ***allocation scheduling*** that deals with the assignment of patients to operating rooms and their sequencing.

More specifically, according to the operating theatre resource management strategy, three different surgery management procedures have been studied [5]:

- *Block scheduling*: A preliminary timetable called master schedule is established in order to allocate time slots to surgeons, groups of surgeons or medical specialties.
- *Open scheduling*: Patients are scheduled without any specialty related restriction.
- *Modified block scheduling*: It is an intermediate strategy that combines the block and open scheduling strategies.

In the literature, the operating room planning and scheduling problems are addressed jointly or separately. In the former case, the two problems are usually considered in a sequential manner using two-phase approaches [5–8]. In the first phase, mathematical formulations are proposed for the ORP problem and in the second phase, heuristic approaches are devised for the ORS problem. In more recent research works, the operating room planning and scheduling problems are addressed simultaneously [9–12]. For a recent review, refer to Cardoen et al. [13] and Francesca and Rosita [14].

The operating room scheduling is a common problem in healthcare sector. Different variants have been investigated, dealing with divers human and material resources (i.e. recovery beds, porters, nurses, etc.) in order to meet specific requirements of the real world healthcare systems. However, this problem is known to be very difficult to solve, especially for real size instances, and most of the papers propose tailored-heuristic approaches.

Guinet and Chaabane [6] develop an extended version of the Hungarian method to assign patients to operating rooms on a medium term. They consider operating rooms and recovery beds. In Fei et al. [5, 7], the daily ORS deals with the availability of operating rooms and recovery beds under respectively open and block scheduling strategies. The problem is modeled as a flow shop-scheduling problem and then solved using a hybrid genetic algorithm. Jebali et al. [8] describe a heuristic approach that solves the ORS problem while considering operating rooms, recovery beds and Intensive Care Unit.

Pariente et al. [15] derive an exact solution for the daily ORS problem in a block scheduling system. They propose a MILP that accounts for the availability of both operating rooms and surgeons, while maximizing the number of treated patients during the planning horizon.

Xiang et al. [4], Guoet et al. [16], and Beliäen and Demeulemeester [17] develop an integrated approach for building nurse and surgery schedules. They model the problem using integer programming and derive solutions using ant colony approach, genetic algorithm and branch-and-price algorithm respectively.

Cardoen et al. [18] consider operation rooms and recovery beds' availability as well as instruments' availability, patients' priority and surgeons' preferences. They formulate the problem as a multi-objective MILP and solve it using a branch-and-price-based approach. In the same vein, Meskens et al. [19] investigate a similar problem where instruments' availability is replaced by surgical team affinities and availability. They model and solve the problem using a constraint programming approach. This same variant of the ORS is studied later by Wang et al. [20]. The authors develop and compare two models using MILP and constraint programming respectively. Augusto et al. [21] study the transfer of patients between hospital beds and operating rooms, the cleaning activities of operating rooms as well as the surgery and the recovery activities. They propose a nonlinear mathematic formulation that they solve using a Lagrangian relaxation-based method. Zhao and Li [22] investigate the operating room availability and the sequence-dependent setup times between surgeries. They describe and compare two programs: a mixed integer nonlinear program (MINLP) and a constraint program.

Recently, stochastic optimization started to emerge to address uncertainty issues. Min and Yih [23] study the elective surgery-scheduling problem with uncertain surgery operations (surgery durations, surgical intensive care unit availability) over multi-periods. They propose a stochastic MILP that minimizes patient costs and expected overtime costs. The problem is then solved using a sampling-based approach. Wang et al. [24] consider uncertain surgery durations and emergency demands. They present an integer-programming problem that minimizes the total expected operating cost including fixed costs related to the number of open operating rooms and expected costs of variable overtime. A column-generation-based heuristic algorithm is devised to solve the problem.

Hans et al. [25] propose constructive and local search heuristics to solve the robust surgery-scheduling problem in block scheduling strategy. The objective is to improve operating room scheduling and reduce the risk of overtime and thus patients' cancellation.

In public healthcare facilities, dynamic generation of patient schedules having various priorities is modelled by Patrick et al. [26] as a Markov decision process. The model is solved using approximate dynamic programming and the solutions quality is analysed through simulation.

Simulation approaches have been investigated for the ORS problem. Persson and Persson [27] address the patient surgery scheduling problem at a hospital in Sweden under specific medical, economic and time constraints. To prevent long queues, patients are allowed to be scheduled at other hospitals according to the Swedish health policy. The authors develop a hybrid simulation and integer-programming approach for solving the problem. Simulation is also used in Patrick and Puterman [28] to improve resource utilization for diagnostic services through flexible inpatient scheduling. López et al. [29] and Prudtikul and Pathomsiri [30] adopt simulation models to analyse the impact of various improvement strategies. Saremi et al. [31] propose three simulation-based tabu search methods for outpatient scheduling in operating theatre with stochastic service times.

To the best of our knowledge, hospital bed capacity constraints are not yet considered when solving ORS problem although they have crucial impact on the number of operated surgery cases and consequently on the hospital profit. However, some research works deal with the hospital bed-planning problem that assigns beds to patients. Wang et al. [32] investigate the bed allocation problem and implement a dynamic dispatching approach. They maximize the hospitalization profit and minimize the bed supplementing cost with both acute and ordinary inpatients. Thompson et al. [33] study a practical bed-planning problem and present an admission decision support system based on finite-horizon Markov decision process model. They implement their system for a hospital ward and prove to bring significant revenue. Ben Bachouch et al. [34] present a decision support tool based on an integer linear model for hospital bed planning with various constraints (incompatibility between pathologies, segregated rooms, continuity of care, etc.). Holm et al. [35]

describe a discrete event simulation approach for the allocation of hospital beds in order to minimize crowding. They also develop optimization algorithms for analyzing the simulation model output based on prevalence and incidence of crowding bed usage.

We notice that the literature pertaining to surgery scheduling focuses on the operating theatre and its human and material resources management. However, there is no previous work that considers the entire surgery process. Moreover, most developed approaches are approximate in nature and literature about exact methods is scarce. In this paper, we build an integrated surgery schedule that takes into account hospital beds, operating rooms and recovery beds. We propose two effective MILP that can be solved to optimality in a reasonable computation time using a commercial software package.

3 Problem statements

The Integrated Elective Surgery-Scheduling Problem consists on sequencing a fixed number of patients and assigning them to resources, over a 1 day period, while considering pre-operative, per-operative and post-operative phases.

In the pre-operative phase, ward admission is performed and patients are assigned to hospital beds. In the per-operative phase, patients are transferred to the operating theatre where they are assigned to operating rooms and then to recovery beds. Finally, in the post-operative phase, patients return to the hospital beds.

A specific feature of the private healthcare sector is that surgeons are independent service providers and are not engaged by the healthcare facility. Consequently, patients choose their surgeon and decide together where and when the surgery will be performed. Thus, when scheduling surgeries, the availability of surgeons and their preferences regarding the day and the time of their surgeries, should be taken into account.

Moreover, in a private healthcare facility, most patients do not need staying for a long period at the hospital ward and thus most of them are outpatients or inpatients requiring only one overnight stay.

Furthermore, the hospital ward is usually composed of single-bed rooms available in a limited number. Consequently, the assignment of patients to hospital beds does not require the integration of the commonly handled incompatibility constraints between man and women, pathologies, etc. Besides, in order to optimize the use of these beds, patients may be assigned to different rooms in the pre-operative and post-operative phases, which allows the use of their rooms by other patients when they are in the operating theatre. However, additional rooms' cleaning and disinfection operations will be required.

In order to optimise the surgery process resources and to shorten the patient's waiting time in the healthcare facility, the patients' average length of stay is minimized. However, an optimal solution may schedule outpatients in the evening and inpatients in the morning, leading more patients to stay overnight in the healthcare facility. Consequently, less beds will be available for the next day activity and surgeries may be cancelled for beds shortage. To overcome this drawback, we investigate the minimization of patients' overnight stays. Indeed, this may schedule outpatients early in the day so less patients are likely to stay overnight in the healthcare facility and thus more free beds will be available for the next day activity.

In this study, we consider small to medium-sized private healthcare facilities where most cases are outpatients. Moreover, we make the following assumptions:

- Emergency cases are not taken into account;
- Human and instruments resources are always available whenever they are needed;
- The operating theatre comprises identical multifunctional rooms without any specialty's restriction. Therefore, the open scheduling strategy is adopted;
- The surgery operation time, the recovery time and the length of stay before and after the surgery are known in advance, although they are subject to uncertainties in practice;
- The surgery operation time comprises the time for preparing the patient (e.g., anaesthesia) in the operating room as well as the surgical act time;
- Each patient is associated to a specific surgeon;
- The availability of each surgeon is known in advance. It is expressed through a time-window during which the surgeon is available at the healthcare facility;
- The operating theatre is open all day (24 h/24 h);
- The hospital ward is composed of single-bed rooms available in a limited number.

This problem can be modelled as a three-stage hybrid flow shop-scheduling problem that uses four different resources (see Fig. 1): Hospital beds, operating rooms, surgeons and recovery beds. Furthermore, the problem exhibits the following additional features:

- *Patients flow recirculation*: patients can be assigned to different hospital rooms in the pre-operative and post-operative phases. Thus, resources of the hospital ward (Stage 1) are used twice through the surgery process;
- *Resource synchronization*: In the operating room (Stage 2), the surgery operation requires two different resources simultaneously: operating rooms and surgeons;

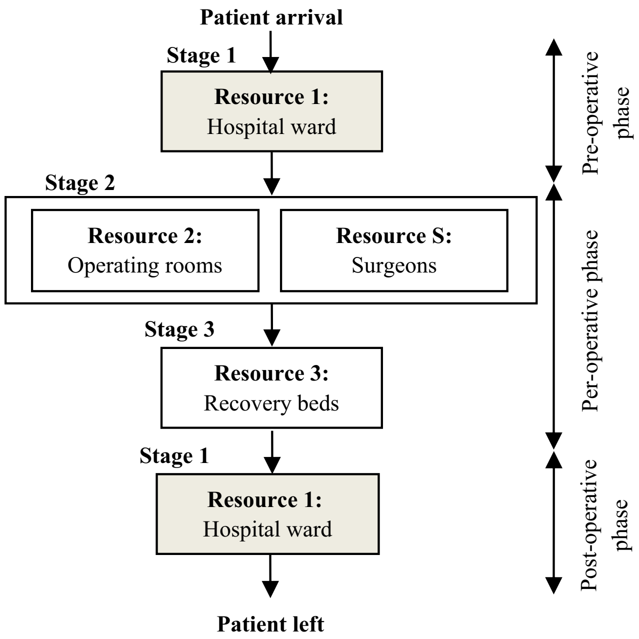


Fig. 1 Patient's flow through the surgery process

- *Dedicated resources*: Each patient is operated by a pre-assigned surgeon who referred her/him to the healthcare facility;
- *Blocking constraints*: There is no buffer between stages. Resources of a stage remain blocked even when the required cares are completed until resources of the next stage become available.

According to [36], the two stage-hybrid flow shop-problem with one machine in the first stage and two or more machines in the second stage is an NP-hard problem in the strong sense. Thus, our problem is NP-hard in the strong sense as well.

Throughout the past three decades, several works have dealt with the hybrid flow shop-scheduling problem, leading to the development of exact and approximate methods. We refer the lecture to [37] for a comprehensive survey.

However, quite a few papers have been interested in the hybrid flow shop-scheduling problem with recirculation. Bertel and Billaut [38] present an industrial scheduling problem that they model as a three-stage hybrid flow shop-problem with recirculation. They develop a liner program and a genetic algorithm for its solution. Boudhar and Meziani [39] consider a two-stage hybrid flow shop-problem with one machine in the first stage and two identical parallel machines in the second stage. Jobs can be recirculated a fixed number of times in the second stage. They propose a linear program and several heuristics to solve it.

Similarly, few researches investigate scheduling problems with dedicated machines. Lin and Liao [40] consider a two-stage hybrid flow shop-problem with sequence-dependent setup times in stage 1, dedicated machines in stage 2 and due date constraints. They develop heuristics that minimize the weighted maximal tardiness time. Yang [41] propose a new complexity proof of the two-stage hybrid flow shop-scheduling problem with dedicated machines in stage 2. He shows that the problem is unary NP-complete.

In this paper, we present two mixed integer linear programs for the IESSP modelled as a three-stage hybrid flow shop-problem with recirculation, resource synchronization, dedicated machines and blocking constraints.

4 MILP formulations

Prior to presenting the two MILP formulations, we introduce the following notation.

4.1 Notation

In order to take into account the recirculation on the hospital ward (Stage 1), we duplicate it into two separate stages denoted by Stage 1 and Stage 4. We also duplicate its resources into two sets: R_1 and R_4 that correspond to hospital rooms assigned before and after surgeries, respectively.

Moreover, for a better understanding of the model, we define an extra stage for the surgeons, denoted by Stage 5.

Sets:

$\boldsymbol{I}$ set of stages indexed by i and i', $\boldsymbol{I} = \{1, 2, \ldots, 5\}$
$\boldsymbol{S}$ set of surgeons, indexed by s
$\boldsymbol{J}$ set of patients, indexed by j and k
$\boldsymbol{J_s}$ set of patients associated to surgeon $s \in \boldsymbol{S}$, $\boldsymbol{J_s} \subset \boldsymbol{J}$
$\boldsymbol{R_i}$ set of available resources at Stage i, indexed by l. We notice here that $\boldsymbol{R_1} = \boldsymbol{R_4}$ and $\boldsymbol{R_5} = \boldsymbol{S}$.

Data:

$\boldsymbol{p_{ij}}$ time to provide care at Stage i to patient j.
$\boldsymbol{Tstart_s}$ and $\boldsymbol{Tfinish_s}$ the earliest start time and the latest finish time for surgeon $s \in \boldsymbol{S}$ to perform surgeries, respectively. They express the availability and/or the time preferences of surgeon $s \in \boldsymbol{S}$. These time limits can be further specified for each patient $j \in J_s$ if necessary. In this case, we define $\boldsymbol{Tstart_{sj}}$ and $\boldsymbol{Tfinish_{sj}}$ as the earliest start time and the latest finish time for surgeon $s \in \boldsymbol{S}$ to operate patient $j \in J_s$, respectively.
$\boldsymbol{M}$ a big positive value.

4.2 Network-flow model

The first model is a network-flow MILP that uses the same variable $\boldsymbol{X}$ to express the decisions related to patient sequencing and assignment to resources. The decision variables are defined as follows:

$\boldsymbol{X_{ijk}}$ equals 1 if in Stage i, a same resource is assigned to patient k immediately after patient j, and 0 otherwise.

$\boldsymbol{Xs_{ilj}}$ equals 1 if patient j is the first patient assigned to resource l of Stage i, and 0 otherwise.

$\boldsymbol{Xt_{ij}}$ equals 1 if patient j is the last patient assigned to a resource in Stage i, and 0 otherwise.

$\boldsymbol{Q^{14}_{jk}}$ equals 1 if a same clinic room is assigned to patient k at Stage 4, after her/his surgery operation, immediately after patient j who is still in Stage 1, and 0 otherwise.

$\boldsymbol{Q^{41}_{jk}}$ equals 1 if a same clinic room is assigned to patient k in Stage 1, before her/his surgery operation, immediately after patient j who is in Stage 4, and 0 otherwise.

$\boldsymbol{T_{ij}}$ expresses the arrival time of patient j at Stage i.

$\boldsymbol{d_{ij}}$ defines the blocking/waiting time of patient j at Stage i.

Using these definitions, the network-flow mixed-integer program (NF_MILP) can be written as follows.

$$\text{NF\_MILP : Minimise } \frac{1}{|J|}\sum_{j}\left(T_{4j}+p_{4j}-T_{1j}\right) \tag{1}$$

Subject to

$$\sum_{l\in R_1} Xs_{1lk} + \sum_{j\in J\backslash\{k\}} X_{1jk} + \sum_{j\in J\backslash\{k\}} Q^{41}_{jk} = 1 \quad \forall\, k\in J \tag{2}$$

$$\sum_{l\in R_4} Xs_{4lk} + \sum_{j\in J\backslash\{k\}} Xs_{4jk} + \sum_{j\in J} Q^{14}_{jk} = 1 \quad \forall k\in J \tag{3}$$

$$\sum_{l\in R_i} Xs_{ilk} + \sum_{j\in J\backslash\{k\}} X_{ijk} = 1 \quad \forall i\in\{2,3\}, k\in J \tag{4}$$

$$Xs_{5sk} + \sum_{j\in J_s\backslash\{k\}} X_{5jk} = 1 \quad \forall s\in S, k\in J_s \tag{5}$$

$$\sum_{k\in J\backslash\{j\}} X_{1jk} + \sum_{k\in J} Q^{14}_{jk} + Xt_{1j} = 1 \quad \forall j\in J \tag{6}$$

$$\sum_{k\in J\backslash\{j\}} X_{4jk} + \sum_{k\in J\backslash\{j\}} Q^{41}_{jk} + Xt_{4j} = 1 \quad \forall j\in J \tag{7}$$

$$\sum_{k\in J\backslash\{j\}} X_{ijk} + Xt_{ij} = 1 \quad \forall i\in\{2,3\}, j\in J \tag{8}$$

$$\sum_{k\in J_s\backslash\{j\}} X_{5jk} + Xt_{5j} = 1 \quad \forall s\in S, j\in J_s \tag{9}$$

$$\sum_{j\in J} Xs_{ilj} \leq 1 \quad \forall i\in I\backslash\{1,4\}, l\in R_i \tag{10}$$

$$\sum_{j\in J} Xs_{1lj} + \sum_{j\in J} Xs_{4lj} \leq 1 \quad \forall\, l\in R_1 \tag{11}$$

$$T_{2j}-T_{1j} = p_{1j}+d_{1j} \quad \forall j\in J \tag{12}$$

$$T_{2j} = T_{5j} \quad \forall\, j\in J \tag{13}$$

$$T_{3j}-T_{2j} = p_{2j}+d_{2j} \quad \forall j\in J \tag{14}$$

$$T_{4j}-T_{3j} = p_{3\mathrm{j}}+d_{3j} \quad \forall j\in J \tag{15}$$

$$M\left(1-Q^{14}_{jk}\right) + T_{4k}-T_{1j} \geq p_{1j} + d_{1j} \quad \forall j\in J, k\in J \tag{16}$$

$$M\left(1-Q^{41}_{jk}\right) + T_{1k}-T_{4j} \geq p_{4j} + d_{4j} \quad \forall j\in J, k\in J\ ,\ \mathrm{j}\neq\mathrm{k} \tag{17}$$

$$M\left(1-X_{ijk}\right) + T_{ik}-T_{ij} \geq p_{ij} + d_{ij} \quad \forall i\in \mathrm{I}\backslash\{5\}, j\in J, k\in J \tag{18}$$

$$M\left(1-X_{5jk}\right) + T_{5k}-T_{5j} \geq p_{5j} \quad \forall j\in J, k\in J\ ,\ \mathrm{j}\neq\mathrm{k} \tag{19}$$

$$T_{5j} \geq Tstart_s \quad \forall s\in S, j\in J_s \tag{20}$$

$$T_{5j} + p_{5j} \leq Tfinish_s \quad \forall s\in S, j\in J_s \tag{21}$$

$$T_{4j} \leq 1440 \quad \forall j\in J \tag{22}$$

$$T_{ij} \geq 0 \quad \forall i\in I, j\in J \tag{23}$$

$$d_{ij} \geq 0 \quad \forall i\in I, j\in J \tag{24}$$

$$X_{ijk}\in\{0,\ 1\} \quad \forall i\in I, j\in J\ ,\ k\in J\ ,\ \mathrm{j}\neq\mathrm{k} \tag{25}$$

$$Xs_{ilj}\in\{0,\ 1\} \quad \forall i\in I, l\in R_i, j\in J \tag{26}$$

$$Xt_{ik}\in\{0,\ 1\} \quad \forall i\in I,\ k\in J \tag{27}$$

$$Q^{14}_{jk}\in\{0,\ 1\} \quad \forall j\in J, k\in J \tag{28}$$

$$Q^{41}_{jk}\in\{0,\ 1\} \quad \forall j\in J, k\in J, \mathrm{j}\neq\mathrm{k}. \tag{29}$$

The objective function (1) minimises the patients' average length of stay in the private healthcare facility. The number of patients' overnight stays minimization will be discussed in Section 4.4.

Equations (2–9) are network-flow conservation constraints and guarantee that all patients are assigned to exactly one resource at each stage (see Figs. 2, 3, and 4).

For the hospital ward, Constraints (2) require that each patient $k\in J$ at the pre-operative phase (Stage 1) is either the first patient assigned to a specific resource $\left(\sum_{l\in R_1} Xs_{1lk} = 1\right)$, or has exactly one predecessor assigned to the same resource. This predecessor can be a patient at the per-operative phase ($\sum_{j\in J\backslash\{k\}} X_{1jk} = 1$) or at the post-operative phase ($\sum_{j\in J\backslash\{k\}} Q^{41}_{jk} = 1$). Constraints (3) impose similar restrictions for patients at the post-operative phase (Stage 4). Indeed, each

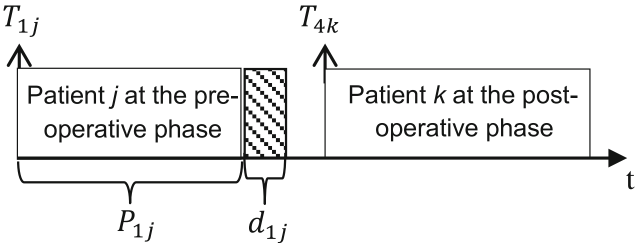

T_{1j}
T_{4k}
Patient j at the pre-operative phase
Patient k at the post-operative phase
t
P_{1j}
d_{1j}

Fig. 2 Patients' schedule at a hospital room: Constraints (16)

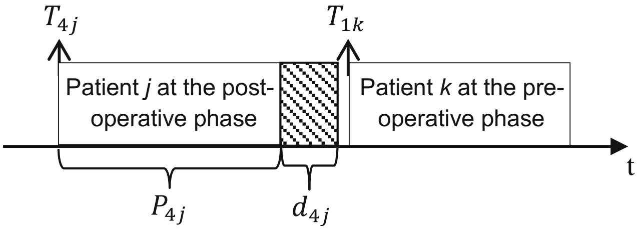

T_{4j}
T_{1k}
Patient *j* at the post-operative phase
Patient *k* at the pre-operative phase
t
P_{4j}
d_{4j}

Fig. 3 Patients' schedule at a hospital room: Constraints (17)

patient $k \in J$ at this stage is the first assigned to a specific resource $\left(\sum_{l\in R_4} Xs_{4lk} = 1\right)$, or has exactly one predecessor assigned to the same resource while being at the per-operative phase ($\sum_{j\in \mathcal{J}\setminus\{k\}} Q^{14}_{jk} = 1$) or at the post-operative phase ($\sum_{j\in \mathcal{J}\setminus\{k\}} X_{4jk} = 1$). Constraints (4) and (5) deal with Stages 2, 3 and 5 where each patient $k \in J$ is either the first assigned to a specific resource or has exactly one predecessor assigned to the same resource.

Similarly, constraints (6–9) ensure that, at each stage, each patient is the last assigned to a resource or has exactly one successor assigned to the same resource.

Constraints (10) and (11) specify that each available resource be assigned to at most one first patient.

Constraints (12–15) link the start time of performing care to a patient at a specific stage, with her/his release time from the previous stage and the corresponding blocking time. Constraints (13) impose resource synchronisation at Stage 2 (operating rooms) and Stage 5 (surgeons) since they must be used simultaneously.

Constraints (16–19) prevent the overlapping of cares performed by a same resource to different patients.

Constraints (20) and (21) express the availability/preferences of each surgeon and Constraints (22) ensure that surgeries are performed in 1 day. Constraints (23–29) define the domains of the decision variables.

Figure 2 illustrates Constraints (16) imposed on the hospital ward (i.e. Stages 1 and 4). It shows that, if a same resource is assigned to patient *k* at the post-operative phase (i.e. Stage 4) immediately after patient *j* at the pre-operative phase (i.e. Stage 1) then variable Q^{14}_{jk} is set to 1 and constraint $T_{4k} - T_{1j} \geq p_{1j} + d_{1j}$ holds.

Similarly, Fig. 3 illustrates Constraints (17). It shows that, if a same resource is assigned to patient *k* at the pre-operative phase (i.e. Stage 1) immediately after patient *j* at the post-operative phase (i.e. Stage 4) then variable Q^{41}_{jk} is set to 1 and constraint $T_{1k} - T_{4j} \geq p_{4j} + d_{4j}$ holds.

In order to better explain patients sequencing and assignment to the hospital ward resources, we consider an example with 3 patients and 2 hospital rooms. Figure 4 presents a Gantt chart that illustrates a feasible patients' schedule satisfying Constraints (2–3); (6–7) and (11).

In this example, non-zero decision variables related to patients' sequencing and assignment in the hospital ward (Stage 1 and Stage 4) have the following values:

$$\text{For room 1}: Xs_{113} = 1, X_{132} = 1, Q^{14}_{23} = 1, Xt_{43} = 1.$$
$$\text{For room 2}: Xs_{422} = 1, Q^{41}_{21} = 1, Q^{14}_{11} = 1, Xt_{41} = 1.$$

Each hospital room is assigned to a sequence of patients and each patient is assigned to one resource at the pre-operative phase and one resource at the post-operative phase.

4.3 Assignment model

The second model is an assignment-based MILP that extends the "standard" mathematical programming formulation of the hybrid flow shop-scheduling problem. Unlike the network-flow model (NF_MILP) where X variables express both patients' sequencing and assignment to resources, the assignment-based model uses two different variables Y and W for these two decisions respectively, defined as follows:

$\boldsymbol{Y^{ii'}_{jk}}$ equals 1 if patient *k* in Stage *i'* is scheduled after patient *j* in Stage *i*, and 0 otherwise where $(i, i') \in \{(1, 1), (2, 2), (3, 3), (4, 4), (5, 5), (1, 4), (4, 1)\}$.

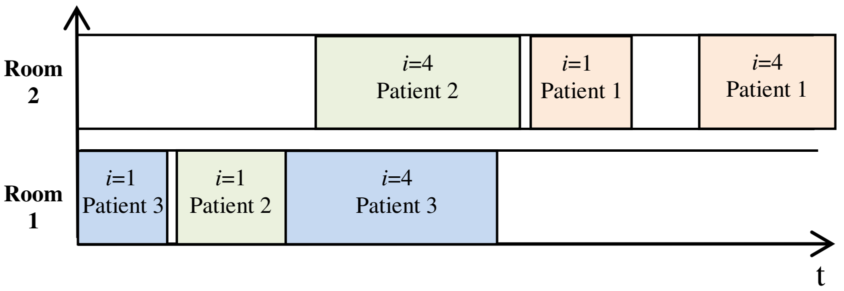
$\boldsymbol{W_{ijl}}$ equals 1 if resource *l* of Stage *i* is assigned to patient *j*, and 0 otherwise

$\boldsymbol{T_{ij}}$ expresses the arrival time of patient *j* at Stage *i*

$\boldsymbol{d_{ij}}$ defines the blocking/waiting time of patient *j* at Stage *i*.

We notice here that X variables require consecutive patients to be served immediately one after the other by a same resource of a stage whereas Y variables are limited to the order of serving two patients in a specific stage. More specifically, $\boldsymbol{X_{ijk}}$ equals one when a same resource is assigned to patient *k* immediately after patient *j* in Stage *i*, while $\boldsymbol{Y^{ii}_{jk}}$ equals one when patient *k* is scheduled after patient *j* (and not necessarily

Fig. 4 Patients' schedule at the hospital ward



immediately after patient j nor being served by a same resource) in Stage i.

Using these definitions, the assignment-based model (A_MILP) can be written as follows.

$$\text{A\_MILP : Minimise} \quad \frac{1}{|J|}\sum_{j}\left(T_{4j}+p_{4j}-T_{1j}\right) \tag{30}$$

Subject to
(12–15), (20–24)

$$\sum_{lR_i} W_{ijl}=1 \quad \forall\, i\in I\backslash\{5\},\ j\in J \tag{31}$$

$$W_{5js}=1 \quad \forall s\in S, j\in J_s \tag{32}$$

$$T_{ik}\geq T_{ij}+p_{ij}+d_{ij}-\mathrm{M}\ \times\left(1-Y_{jk}^{ii}\right)-\mathrm{M}\times\left(1-W_{ijl}\right)-\mathrm{M}\times\ \left(1-W_{ikl}\right)\quad \forall i\in I\backslash\{5\},\ l\in R_i\ ,\ j\in J, k\in J, j\neq k \tag{33}$$

$$T_{5k}\geq T_{5j}+p_{5j}-\mathrm{M}\times\left(1-Y_{jk}^{55}\right)\quad \forall\ s\in S, j\in J_s, k\in J_s, j\neq k \tag{34}$$

$$T_{1k}\geq T_{4j}+p_{4j}+d_{4j}-\mathrm{M}\times\ \left(1-Y_{jk}^{41}\right)\ -\mathrm{M}\times\left(1-W_{1kl}\right)\ -\mathrm{M}\times\left(1-W_{4jl}\right)\quad \forall\ l\in R_1, j\in J, k\in J, j\neq k \tag{35}$$

$$T_{4k}\geq T_{1j}+p_{1j}+d_{1j}-\mathrm{M}\times\ \left(1-Y_{jk}^{14}\right)\ -\mathrm{M}\times\left(1-W_{1kl}\right)-\mathrm{M}\times\left(1-W_{4jl}\right)\quad \forall\ l\in R_1,\ j\in J, k\in J \tag{36}$$

$$Y_{jk}^{ii}+Y_{kj}^{ii}=1 \quad \forall i\in I,\ j\in J,\ k\in J, j\neq k \tag{37}$$

$$Y_{jk}^{14}+Y_{kj}^{41}=1 \quad \forall i\in I,\ j\in J, k\in J, j\neq k \tag{38}$$

$$W_{ijl}\in\{0,\ 1\} \quad \forall i\in I, l\in R_i, j\in J \tag{39}$$

$$Y_{jk}^{ii'}\in\{0,\ 1\} \quad \forall j\in J, k\in J,\ i\in I,\ i'\in I, j\neq k. \tag{40}$$

The objective function (30) is the same as (1). It minimises the patients' average length of stay in the private healthcare facility. Constraints (12–15) and (20–24) are those defined for NF_MILP model.

Constraints (31) and (32) ensure that at each stage, each patient is assigned to exactly one resource.

Constraints (33–36) are similar to Constraints (16–19) and prevent the overlapping of cares performed by a same resource to any two patients.

Constraints (37) and (38) express the sequencing constraints between each pair of patients at a same stage and expressions (39) and (40) are the integrality constraints to be imposed on the variables.

4.4 Extensions

In this section, we present some possible extensions to the proposed models that cope with further realistic features.

4.4.1 Minimizing the patients overnight stays

In order to have enough free hospital beds for the next day activity, the number of patients who spend the night in the hospital should be minimised.

In practice, outpatients have to leave the hospital ward before a fixed time limit T_{limo} (e.g., 7 pm) on the surgery day, whereas the inpatients spend at least one night at the hospital and have to leave the ward before a fixed time limit T_{limi} (e.g., noon) of the hospital discharge day. To account for this feature, we denote by J_o the set of outpatients and by J_i the set of inpatients.

The minimization of the patients' overnight stays requires considering the objective function (41) and Constraints (42–44) in both models NF_MILP and A_MILP, where $\boldsymbol{N_j}$ is a binary variable that equals 1 if outpatient j spends a night at the hospital ward or inpatient j spends more than one night at the hospital ward, and 0 otherwise.

$$\text{Minimize}\sum_{j\in J} N_j \tag{41}$$

$$T_{4j}+p_{4j}\leq N_j\ \times\mathrm{M}+T_{limo} \quad \forall j\in J_o \tag{42}$$

$$T_{4j}+p_{4j}\leq N_j \times \mathrm{M}+T_{limi} \quad \forall j\in J_i \tag{43}$$

$$N_j\in\{0,\ 1\} \quad \forall j\in J \tag{44}$$

4.4.2 Setup time sequence-dependent between surgeries

After each surgery, the operating room should be cleaned, disinfected and sterilized for the next surgery. Moreover, some surgeries need specific room preparation (e.g., installing specific equipment). These setup operations may depend on the sequence of surgery types.

To cope with this situation, for Stage 2 (operating rooms), Constraints (18) should be replaced by Constraints (45) in NF_MILP and Constraints (33) should be replaced by Constraints (46) in A_MILP.

$$M\times\left(1-X_{2jk}\right)+T_{2k}-T_{2j}\geq p_{2j}+d_{2j}+\alpha_{jk} \quad \forall j\in J, k\in J, j\neq k \tag{45}$$

$$\mathrm{T_{2k}}\geq \mathrm{T_{2j}}+\mathrm{p_{2j}}+\mathrm{d_{2j}}+\alpha_{\mathrm{jk}}-\mathrm{M}\times\left(1-\mathrm{Y}^{22}_{\mathrm{jk}}\right)-\mathrm{M}\times\left(1-\mathrm{W_{2jl}}\right)-\mathrm{M}\times\left(1-\mathrm{W_{2kl}}\right) \quad \forall l\in R_2, j\in J, k\in J,\ j\neq k \tag{46}$$

where α_{jk} is the setup time when the surgery of patient k is performed immediately after the surgery of patient j.

4.4.3 Allowing patient recovery in the operating room

At the end of the surgery operation, the patient is transferred to a recovery bed to wake up slowly under appropriate personal and equipment monitoring. However, if no recovery bed is available, the patient can wake up in the operating room until the discharge of a recovery bed or until she/he is transported back to the ward after her/his recovery.

To cope with this situation, Constraints (14) and (15) are replaced by Constraints (47) and (48) in both models NF_MILP and A_MILP. Also, for Stage 2 (operating rooms), Constraints (18) and (19) in NF_MILP are replaced by Constraints (49) and (50), respectively. Similarly, in A_MILP, Constraints (32) and (33) are replaced by Constraints (51) and (52), respectively. In addition, we append Constraints (53) to both models to limit the recovery in the operating room to its estimated time.

$$T_{3j}-T_{2j}=p_{2j}+d_{2j}+r_j \quad \forall j\in J \tag{47}$$

$$T_{4j}-T_{3j}=p_{3\mathrm{j}}+d_{3j}-r_j \quad \forall j\in J \tag{48}$$

$$\mathrm{M}\left(1-X_{2jk}\right)+T_{2k}-T_{2j}\geq\ p_{2j}+d_{2j}+r_j \quad \forall j\in J, k\in J, j\neq k \tag{49}$$

$$\mathrm{M}\left(1-X_{3jk}\right)+T_{3k}-T_{3j}\geq\ p_{3j}+d_{3j}-r_j \quad \forall j\in J, k\in J, j\neq k \tag{50}$$

$$T_{2k}\geq T_{2j}\ +\ p_{2j}+r_j-\mathrm{M}\ \times\left(1-Y^{22}_{jk}\right)-\mathrm{M}\times\left(1-W_{2jl}\right)-\mathrm{M}\times\left(1-W_{2kl}\right) \quad \forall l\in R_2,\ j\in J, k\in J, j\neq k \tag{51}$$

$$T_{3k}\geq T_{3j}\ +\ p_{3j}-r_j-\mathrm{M}\ \times\left(1-Y^{33}_{jk}\right)-\mathrm{M}\times\left(1-W_{3jl}\right)-\mathrm{M}\times\left(1-W_{3kl}\right) \quad \forall l\in R_3,\ j\in J, k\in J, j\neq k \tag{52}$$

$$0\leq\ r_j\leq p_{3j} \quad \forall j\in J \tag{53}$$

where r_j is a decision variable that indicates the recovery time spent by patient j in the operating room.

4.4.4 Performing the acts of the same surgeon successively

Surgeons may impose performing their surgeries successively to reduce their travel time to/from the healthcare facility as well as predictive errors in case durations and tardiness from scheduled start times. This requires scheduling the cases of a same surgeon in sequence [42, 43]. To deal with this situation, Constraints (19) are replaced by Constraints (54) in NF_MILP and Constraints (34) are replaced by Constraints (55) in A_MILP.

$$T_{5k}X_{5jk}\ =\left(T_{5j}+p_{5j}\right)X_{5jk} \quad \forall s\in S, j\in J_s, k\in J_s, j\neq k \tag{54}$$

$$T_{5k}Y^{55}_{jk}=\left(T_{5j}+p_{5j}\right)Y^{55}_{jk} \quad \forall s\in S, j\in J_s, k\in J_s, j\neq k \tag{55}$$

These nonlinear constraints can be linearized using the Reformulation Linearization Technique developed by Sherali and Adams in 1990 and 1994 [44, 45].

5 Computational experiments

To evaluate the performance of the proposed NF-MILP and A_MILP models, we conducted experiments on real-world data and on a set of randomly generated instances. Moreover, we investigated the minimization of two criteria: patients' average length of stay and patients' overnight stays.

The proposed models were solved using CPLEX 12.2 solver with the default setting. All the experiments were run on an Intel PC (R) Xeon (R) with a 3.30 GHz CPU processor and 8 GB RAM.

In the following paragraphs, we present the test-bed instances, then we discuss the computational effort required to solve them optimally. Finally, we investigate the impact of the objective function on the solution's quality.

5.1 Test-bed instances

The computational study is carried out on two types of instances: (1) real-world instances provided by a Tunisian private clinic: *Clinique Ennasr* and (2) randomly generated instances.

5.1.1 Real-world instances provided by Clinique Ennasr

Clinique Ennasr is a private Tunisian clinic that is located in Tunis. It offers a range of clinical services corresponding to different medical and surgical specialties with a major on oncology. Its operating theatre comprises four identical operating rooms and four recovery beds. It employs a qualified nursing staff that along with well-dimensioned surgical facilities and instruments, allows the opening of the operating theatre all day long (24 h/24 h).

Clinique Ennasr provided us with six real-data instances that represent different surgical activity levels in 2014. Table 1 presents the main characteristics of these instances: Column 1 provides the instance number (*Inst.*). Column 2 gives the number of patients (# *Patients*). Column 3 indicates the number of surgeons (# *Surgeons*). The last column presents the number of patients per clinic room ($\frac{\# \textit{Patients}}{\# \textit{Clinic rooms}}$).

Table 1 Real-world instances of *Clinique Ennasr*

Inst.	# Patients	# Surgeons	$\frac{\# \text{Patients}}{\# \text{Clinic rooms}}$
I1	20	10	1
I2	20	13	1
I3	23	12	1.15
I4	23	10	1.15
I5	25	13	1.25
I6	29	13	1.45

Real-world instance data specifies for each surgery case, the surgery date, the referred surgeon, the surgery specialty as well as pre-operative, surgery, recovery and post-operative times. This test-bed involves 20 clinic rooms that are available at the beginning of each day, on average.

We notice here that on a given day, each surgeon operates 1.9 cases on average. This reflects the real-world context of *Clinique Ennasr* and its private surgeons. Indeed, the clinic, as a private facility, collaborates with various surgeons of different specialties and surgeons as independent service provider conduct their surgeries in several healthcare facilities.

5.1.2 Generated data

In order to conduct further experiments, we generated two sets of 60 instances. The first set mimics the surgical activity of *Clinique Ennasr* while the second set simulates the activity of a larger surgery unit (see Table 2).

Table 2 presents, for each set of instances, the number of generated instances (# *Instances*), the number of operating rooms (# *Operating rooms*), the number of recovery beds (# *Recovery beds*) and the number of clinic rooms (# *Clinic rooms*).

In set 1, according to the Tunisian regulation that imposes to have at least as many recovery beds as operating rooms [46], the number of recovery beds is set to the number of operating rooms. However, in the state of the art, it is often recommended to have a number of recovery beds equal to 1.5 times the number of operating rooms [47], which is considered in set 2.

For each set of instances, we simulated various activity levels of the facility that correspond to different numbers of surgeries to be scheduled (see Table 3). Moreover, in order to draw the surgeons' activity on a given day, we assigned randomly three surgery cases on average to each surgeon. However, we do not impose any preference/availability restriction for the surgeons.

For each subset, we generated five instances with different processing times (per-operative times, operating times,

Table 2 Description of the surgical unit of the generated instances

Set of instances	# Instances	# Operating rooms	# Recovery beds	# Clinic rooms
1	25	4	4	20
2	35	6	9	30

Table 3 Main characteristics of the generated subsets of instances

Inst. subset	# Patients	# Surgeons	$\frac{\#\ \text{Patients}}{\#\ \text{Clinic rooms}}$
I-1-24	24	8	1.20
I-1-26	26	9	1.30
I-1-28	28	9	1.40
I-1-30	30	10	1.50
I-1-32	32	11	1.60
I-2-32	32	11	1.07
I-2-34	34	11	1.13
I-2-36	36	12	1.20
I-2-38	38	13	1.27
I-2-40	40	13	1.33
I-2-42	42	14	1.40
I-2-44	44	15	1.47

recovery times and post-operative times). To estimate these times, we conducted an experimental study on historical data related to the surgery activity during 2014 at *Clinique Ennasr*. The results indicate that the processing times depend on three types of elective surgery cases:

- Outpatients who can leave the hospital after their recovery, without overnight staying. They represent 40 % of the patients.
- Medium length of stay inpatients who are required to spend only one night in the clinic. They represent 40 % of the patients.
- Long length of stay inpatients who need to spend more than one night in the clinic. They represent 20 % of the patients.

According to the experimental study, we set the per-operative times to 2 h for outpatients and medium length of stay inpatients. However, for long length of stay inpatients, they usually enter the clinic a day before the surgery, and thus their per-operative time is set to only 10 min.

For surgery and post-operative times, we analysed the data using Minitab software [48] and we compared the P_value of different distributions functions. Table 4 presents the obtained results for the four best fitting distributions functions: Normal, Exponential, Lognormal and Gamma, for each patient type.

Table 4 shows that the best fitting probability distribution function is the lognormal distribution since it has the largest P_value exceeding the significance level α of 5 %. This result confirms previous works, where the surgery times are usually generated using a lognormal distribution ([8, 22, 49, 50]) or a Pearson III distribution [6].

Tables 5 and 6 provide the mean and the standard deviation of the considered lognormal distribution for surgery times and post-operative times, respectively.

Table 4 P-value Results

		Outpatient	Medium length of stay in-patients	Long length of stay in-patients
surgery time	Normal	0.08	0.025	0.699
	Exponential	0.021	0.055	0.003
	Lognormal	0. 81	0.28	0.78
	Gamma	>0.250	>0.250	>0.250
Post-operative time	Normal	0.006	0.011	0.052
	Exponential	0.088	<0.003	0.306
	Lognormal	0.25	0.13	0.37
	Gamma	0.062	0.032	>0.250

Moreover, in order to better estimate surgery and post-operative times, we propose to use a truncated lognormal distribution that imposes minimum and maximum values.

To validate the characteristics of the selected distribution functions, we presented them to the practicians of *Clinique Ennasr* who suggest minor rounding adjustments. The final characteristics of the truncated lognormal distribution for the surgery and post-operative times are displayed in Tables 7 and 8, respectively. The columns give the mean, the standard deviation, the minimum and the maximum values in minutes, respectively,

For recovery times, we set them equal to surgery times for all inpatients and to surgery times minus 10 min for outpatients.

5.2 Computational results

To assess the empirical performance of the proposed formulations NF_MILP and A_MILP, we compared their computational time to solve the test-bed instances optimally.

We conducted three sets of experiments with different objective functions. The first set considers the minimization of patients' average length of stay and the second set minimises the number of patients' overnight stays. Finally, in the third set, patients' average length of stay of patients is minimised with additional constraints on the maximal number of patients' overnight stays.

Table 5 Characteristics of the lognormal distribution for surgery times

Patient type	Mean (min)	Standard deviation (min)
Outpatient	32	21
Medium length of stay in-patients	74	36
Long length of stay in-patients	166	64

Table 6 Characteristics of the lognormal distribution for post-operative times

Patient type	Mean (min)	Standard deviation (min)
Outpatient	306	135
Medium length of stay in-patients	865	785
Long length of stay in-patients	2327	1779

In all our experiments, we pre-set the CPU time limit to 1 h.

In the subsequent paragraphs, the generated instances are denoted by *I-s-n-k* where *s* is the instance set number ($s\in\{1, 2\}$), *n* is the number of patients and *k* is the instance number ($k\in\{1, \ldots, 5\}$).

5.2.1 The patients' average length of stay minimization

In this set of experiments, the objective function of the Integrated Elective Surgeries' Scheduling Problem (IESSP) minimises patients' average length of stay. Tables 9 and 10 display the results of real-world instances and generated instances, respectively. The two first columns indicate the problem instance identification (*Inst.*) and the optimal patients' average length of stay, in minutes (*ALS** (*min*)). The next two columns provide the CPU times, in seconds required by NF_MILP (*CPU_NF* (*s*)) and A_MILP (*CPU_A* (*s*)) formulations, respectively. The last column indicates the CPU time improvement achieved by A_MILP against NF_MILP in percentage (*CPU_IMP* (%)) given by (56).

$$CPU\_IMP(\%) = \frac{CPU\_NF{-}CPU\_A}{CPU_{NF}} \times 100 \qquad (56)$$

*** indicates that no optimal solution was found within 1 h of computation time.

Tables 9 and 10 show that A_MILP model outperforms NF_MILP model. In Table 9, all the real-world instances are optimally solved in less than 21 min. NF_MILP model requires longer CPU times than A_MILP model, ranging from 8.07 s to 1208.92 s, whereas the CPU times of A_MILP model do not exceed 69.44 s. For the generated instances, Table 10 show that, within 1 h of CPU time, A_MILP model provides optimal solutions for all instances in less than 183.71 s in average, whereas NF_MILP model solves only 56.7 % of the instances within 361.10s in average.

Table 7 Characteristics of the selected truncated lognormal distribution for surgery times

Patient type	Mean (min)	Standard deviation (min)	Minimum (min)	Maximum (min)
Outpatient	30	20	15	120
Medium length of stay in-patients	70	35	30	300
Long length of stay in-patients	150	70	60	540

Table 8 Characteristics of the selected truncated lognormal distribution for post-operative times

Patient type	Mean (min)	Standard deviation (min)	Minimum (min)	Maximum (*min*)
Outpatient	300	150	60	480
Medium length of stay in-patients	900	800	600	1740
Long length of stay in-patients	2280	1740	1740	7200

It is worth mentioning that for the generated instances of set 2, and when the number of patients per clinic room is less than 1.2, NF_MILP model requires less CPU time than A_MILP model.

Not surprisingly, we observe that the CPU time increases as the number of patients and the number of patients per clinic room increase. Indeed, the problem becomes more constrained and thus more difficult to solve.

5.2.2 Number of patients' overnight stays minimization

In this second set of experiments, the objective function of IESSP aims to minimise the number of patients' overnight stays. Thus, we replace the objective function by expression (41) and append Constraints (42–44) in both models NF-MILP and A_MILP.

Tables 11 and 12 provide the results of real-world instances and generated instances, respectively. The two first columns indicate the problem instance identification (*Inst.*) and the optimal number of patients' overnight stays (# *PO**). The next two columns provide the CPU times, in seconds required by NF_MILP (*CPU_NF* (*s*)) and A_MILP (*CPU_A* (*s*)) formulations, respectively. The last column indicates the CPU time improvement achieved by A_MILP against NF_MILP in percentage (*CPU_IMP* (%)) and given by (56).

Table 9 Patients' average length of stay minimisation for the real-world instances

Inst.	ALS* (min)	CPU_NF (s)	CPU_A (s)	CPU_IMP (%)
I1	1522.65	8.07	15.80	−95.79
I2	2051.50	10.01	5.21	47.95
I3	1683.30	20.01	12.60	37.03
I4	1525.57	30.01	29.76	0.83
I5	1358.48	30.02	18.50	38.37
I6	1585.66	1208.92	69.44	94.26

Table 10 Patients' average length of stay minimisation for the generated instances

Inst.	ALS* (min)	CPU_NF (s)	CPU_A (s)	CPU_IMP (%)
I-1-24-1	1705.38	120.54	30.56	97.88
I-1-24-2	1751.21	37.47	30.92	17.48
I-1-24-3	1383.17	15.27	34.91	−128.62
I-1-24-4	1485.54	543.76	31.07	94.29
I-1-24-5	1522.80	95.07	28.61	69.91
I-1-26-1	1753.92	634.49	44.40	93.00
I-1-26-2	1630.77	1039.32	43.00	95.86
I-1-26-3	1521.46	398.91	37.62	90.57
I-1-26-4	1618.35	1899.11	84.28	95.56
I-1-26-5	2057.69	856.74	39.72	95.36
I-1-28-1	1709.04	***	53.07	***
I-1-28-2	1576.75	2630.37	54.24	97.94
I-1-28-3	1462.36	358.89	53.76	85.02
I-1-28-4	1536.75	***	425.28	***
I-1-28-5	1838.75	***	6.91	***
I-1-30-1	1534.93	***	230.28	***
I-1-30-2	1652.40	***	83.38	***
I-1-30-3	1782.10	***	90.54	***
I-1-30-4	1673.60	***	100.27	***
I-1-30-5	1582.63	***	90.40	***
I-1-32-1	1594.50	***	301.44	***
I-1-32-2	1475.63	***	171.98	***
I-1-32-3	1602.63	***	302.44	***
I-1-32-4	1621.50	***	197.43	***
I-1-32-5	1646.38	***	642.59	***
I-2-32-1	1511.19	35.24	61.49	−74.49
I-2-32-2	1796.56	68.29	15.13	77.84
I-2-32-3	1662.94	30.27	134.07	−342.91
I-2-32-4	1612.75	37.10	140.26	−278.06
I-2-32-5	1648.06	32.81	89.93	−174.09
I-2-34-1	1930.88	45,00	155.36	−245.24
I-2-34-2	1637.79	56.60	152.87	−170.09
I-2-34-3	1546.62	47.31	66.24	−40.01
I-2-34-4	2107.18	101.70	66.22	34.89
I-2-34-5	1984.29	127.34	142.63	−12.01
I-2-36-1	2010.36	60.91	181.98	−198.77
I-2-36-2	1868.11	69.22	186.44	−169.34
I-2-36-3	1835.81	249.41	205.73	17.51
I-2-36-4	1929.86	51.25	86.77	−69.31
I-2-36-5	1765.92	57.06	189.48	−232.07
I-2-38-1	1578.18	176.02	207.58	−17.93
I-2-38-2	2028.24	81.33	333.35	−309.87
I-2-38-3	1800.32	77.75	31.49	59.50
I-2-38-4	1561.61	550.93	268.31	51.30
I-2-38-5	1503.68	110.73	94.20	14.93
I-2-40-1	1711.35	1147.99	327.84	71.44
I-2-40-2	1829.30	433.44	242.30	44.10
I-2-40-3	2017.63	***	274.43	***
I-2-40-4	1719.60	***	315.04	***
I-2-40-5	1875.18	***	294.50	***
I-2-42-1	1724.40	***	323.98	***
I-2-42-2	1972.48	***	122.37	***
I-2-42-3	1821.43	***	163.80	***
I-2-42-4	1697.19	***	480.33	***
I-2-42-5	1838.17	***	315.03	***
I-2-44-1	1810.43	***	465.49	***
I-2-44-2	1726.66	***	495.14	***
I-2-44-3	1940.41	***	376.62	***
I-2-44-4	1787.09	***	449.26	***
I-2-44-5	1729.50	***	357.855	***

Tables 11 and 12 confirm that A_MILP model consistently outperforms NF_MILP model. Indeed, A_MILP model solves to optimality all the real-world instances in less than 133.52 s of computation time, whereas NF_MILP solves only 66.7 % of these instances within the pre-set CPU time of 1 h. Moreover, NF_MILP model requires longer CPU times than A_MILP model except for the first instance.

For the generated instances, Table 12 show that, within 1 h of computation time, A_MILP model provides optimal solutions to 91.7 % of the instances while requiring 313.01 s of CPU time in average, whereas NF_MILP model solves only 21.7 % of the instances within 1679s of CPU time in average.

Besides, for the instances solved to optimality by both models, A_MILP provides solutions in less CPU time than NF_MILP for 82.3 % of the instances with 91.5 % of CPU time improvement in average.

Finally, here again, we observe that the CPU time increases as the number of patients and the number of patients per clinic room increase. More specifically, instances where the number of patients per clinic room exceeds 1.2 seem to be harder to solve by NF_MILP model.

Table 11 Number of patients' overnight stays minimization for real-world instances

Inst.	# PO*	CPU_N (s)	CPU_A (s)	CPU_IMP (%)
I1	9	11.58	17.35	−49,83
I2	8	175.04	5.04	97.12
I3	7	3167.83	9.88	99.69
I4	6	1757.45	7.96	99.55
I5	5	***	10.73	***
I6	6	***	133.52	***

Table 12 Number of patients' overnight stays minimization for the real-world instances

Inst.	# PO*	CPU_NF (s)	CPU_A (s)	CPU_IMP (%)
I-1-24-1	9	***	16.18	***
I-1-24-2	8	***	15.30	***
I-1-24-3	6	***	17.75	***
I-1-24-4	7	***	34.63	***
I-1-24-5	7	***	37.32	***
I-1-26-1	8	***	49.87	***
I-1-26-2	11	***	7.18	***
I-1-26-3	10	***	55.52	***
I-1-26-4	9	***	48.45	***
I-1-26-5	11	***	44.96	***
I-1-28-1	8	***	288.4	***
I-1-28-2	11	***	232.66	***
I-1-28-3	10	***	74.21	***
I-1-28-4	10	***	425.28	***
I-1-28-5	12	***	67.7	***
I-1-30-1	11	***	131.76	***
I-1-30-2	***	***	***	***
I-1-30-3	11	***	126.46	***
I-1-30-4	***	***	***	***
I-1-30-5	12	***	3044.28	***
I-1-32-1	11	***	832.70	***
I-1-32-2	9	***	980.91	***
I-1-32-3	11	***	364.12	***
I-1-32-4	***	***	***	***
I-1-32-5	***	***	***	***
I-2-32-1	11	226.00	41.12	81.81
I-2-32-2	8	3279.75	151.01	95.40
I-2-32-3	10	28.89	109.44	−278.82
I-2-32-4	10	393.23	42.32	89.24
I-2-32-5	11	292.66	125.56	57.10
I-2-34-1	13	761.46	49.31	93.52
I-2-34-2	12	3077.31	73.64	97.61
I-2-34-3	10	***	128.56	***
I-2-34-4	14	3204.27	156.94	95.10
I-2-34-5	15	43.47	58.34	−34.21
I-2-36-1	15	1859.97	185.63	90.02
I-2-36-2	13	2238.14	148.94	93.35
I-2-36-3	14	2974.97	74.81	97.49
I-2-36-4	11	***	221.04	***
I-2-36-5	13	3456.24	210.17	93.92
I-2-38-1	13	***	269.16	***
I-2-38-2	12	***	256.49	***
I-2-38-3	12	***	245.03	***
I-2-38-4	12	***	213.43	***
I-2-38-5	13	***	915.52	***
I-2-40-1	16	***	325.02	***
I-2-40-2	13	***	434.15	***
I-2-40-3	14	***	314.86	***
I-2-40-4	13	***	442.08	***
I-2-40-5	17	***	928.68	***
I-2-42-1	14	***	387.94	***
I-2-42-2	15	***	395.1	***
I-2-42-3	18	***	389.46	***
I-2-42-4	13	***	545.42	***
I-2-42-5	13	***	922.87	***
I-2-44-1	13	***	408.24	***
I-2-44-2	13	***	2.73	***
I-2-44-3	13	***	524.52	***
I-2-44-4	15	***	621.92	***
I-2-44-5	***	***	***	***

5.2.3 Patients' average length of stay minimization under patients' overnight stays constraints

In this last set of experiments, we consider the minimization of both criteria: patients' average length of stay and number of patients' overnight stays. We apply a hierarchical approach to solve this multi-objective problem.

Since A_MILP model performs better than NF_MILP model, according to the two previous experiments, we use it for the multi-objective problem. Thus, we build the so-called MO_A_MILP model by appending to A_MILP model Constraints (42–44) as well as the following constraint (57),

$$\sum_{j} N_j = OP^* \tag{57}$$

where *OP** is the minimum number of patients' overnight stays.

Constraint (57) enforces the number of patients' overnight stays to be equal to its minimum value *OP**.

Tables 13 and 14 display the computational results of the multi-objective model MO_A_MILP for the real-world instances and generated instances, respectively. The first column indicates the problem instance identification (*Inst.*). The two next columns give the optimal patients' average length of stay

Table 13 Multi-objective solutions for the real-world instances

Inst.	MO_ALS* (min)	MO_CPU (s)	GAP_ALS (%)
I1	1522.80	2.33	−0.009
I2	2051.50	2.80	0
I3	1683.31	7.91	−0.001
I4	1525.57	13.68	0
I5	1358.48	17.79	0
I6	1585.66	53.57	0

Table 14 Multi-objective solutions for the generated instances

Inst.	MO_ALS* (min)	MO_CPU (s)	GAP_ALS (%)
I-1-24-1	1705.38	16.76	0
I-1-24-2	1751.21	37.68	0
I-1-24-3	1383.17	32.78	0
I-1-24-4	1485.54	3.52	0
I-1-24-5	1522.80	23.33	0
I-1-26-1	1753.92	39.78	0
I-1-26-2	1630.77	38.46	0
I-1-26-3	1521.46	10.82	0
I-1-26-4	1618.35	77.76	0
I-1-26-5	2057.69	13.76	0
I-1-28-1	1709.04	43.71	0
I-1-28-2	1576.75	58.52	0
I-1-28-3	1462.36	61.43	0
I-1-28-4	***	***	***
I-1-28-5	1838.75	49.67	0
I-1-30-1	***	***	***
I-1-30-2	+++	+++	+++
I-1-30-3	1782.10	99.89	0
I-1-30-4	+++	+++	+++
I-1-30-5	1582.63	362.72	0
I-1-32-1	1594.50	259.31	0
I-1-32-2	1475.63	562.45	0
I-1-32-3	***	***	***
I-1-32-4	+++	+++	+++
I-1-32-5	+++	+++	+++
I-2-32-1	1511.19	56.36	0
I-2-32-2	1796.63	13.57	−0.003
I-2-32-3	1662.94	34.84	0
I-2-32-4	1612.75	13.47	0
I-2-32-5	1649.94	77.66	−0.11
I-2-34-1	1930.88	54.69	0
I-2-34-2	1637.79	64.46	0
I-2-34-3	1546.62	39.62	0
I-2-34-4	2107.18	162.95	0
I-2-34-5	1984.29	62.64	0
I-2-36-1	2010.36	63.27	0
I-2-36-2	1868.11	56.88	0
I-2-36-3	1835.81	169.43	0
I-2-36-4	1929.86	72.39	0
I-2-36-5	1765.92	21.14	0
I-2-38-1	1578.18	63.58	0
I-2-38-2	2028.24	233.53	0
I-2-38-3	1800.32	181.68	0
I-2-38-4	1561.61	206.72	0
I-2-38-5	1503.68	215.10	0
I-2-40-1	1711.35	329.58	0
I-2-40-2	1829.30	25.70	0
I-2-40-3	2017.63	200.47	0
I-2-40-4	1719.60	274.12	0
I-2-40-5	1875.18	287.56	0
I-2-42-1	1724.40	310.09	0
I-2-42-2	1972.48	110.64	0
I-2-42-3	1821.43	311.64	0
I-2-42-4	1697.19	310.74	0
I-2-42-5	1838.17	408.24	0
I-2-44-1	***	***	***
I-2-44-2	1728.60	524.52	−0.11
I-2-44-3	***	***	***
I-2-44-4	***	***	***
I-2-44-5	+++	+++	+++

in minutes (*MO_ALS** (*min*)) and the CPU times in seconds (*MO_CPU* (*s*)). The last column indicates the gap between the patients' average length of stay (*MO_ALS**) obtained by the multi-objective problem MO_A_MILP and its minimum value (*ALS**) provided by the A_MILP model.

$$GAP\_ALS\ (\%) = \frac{ALS^{*} - MO\_ALS^{*}}{ALS^{*}} \times 100 \qquad (58)$$

In Tables 13 and 14, +++ indicates instances for which the minimum number of patients' overnight stays *OP** is unknown; and *** indicates unsolved instances within 1 h of computation time.

Tables 13 and 14 provide empirical evidence that A_MILP based models perform very well. Table 13 shows that all the real-world instances are optimally solved within less than 53.75 s. For the generated instances, the results of Table 14 indicate that, within the pre-set CPU time limit of 1 h, MO_A_MILP model provides optimal solutions to 81.7 % of the instances while requiring a CPU time of 140.27 s in average, ranging from 36.67 s to 524.52 s. Moreover, the solution of 8.3 % of the instances is not considered because we do not have the minimum number of patients' overnight stays *OP**.

Besides, appending constraints on the number of patients' overnight stays has a little impact on the patients' average length of stay. Indeed, for 89.8 % of the generated instances, patients' average length of stay (*MO_ALS**) provided by the hierarchical approach is equal to its minimum value (*ALS**). For the other instances, patients' average length of stay is increased by only 0.04 % in average.

6 Conclusions and future research

In this work, we focused on the optimal solution of the Integrated Elective Surgeries' Scheduling Problem

(IESSP) that arises in a privately operated healthcare facility. It consists on scheduling surgery cases over 1-day period while taking into account four main resources: Hospital beds, operating rooms, surgeons and recovery beds.

We modelled this problem as a three-stage hybrid flow-shop scheduling problem with recirculation, resource synchronization, dedicated machines, and blocking constraints. We developed two mixed integer linear programs that can be efficiently solved using a general-purpose MILP solver. The first program is based on network flow model (NF_MILP) and the second program is an assignment-based model (A_MILP). Both models are compact and can be easily extended to accommodate several specific restrictions of the real-world healthcare context.

Two objective functions were investigated: patients' average length of stay and number of patients' overnight stays minimization. First, we tested the two criteria separately and then we considered them simultaneously using a hierarchical approach.

The results of the computational experiments conducted on six real-world instances delivered by *Clinique Ennasr* and 60 instances generated randomly provide an empirical evidence that the assignment-based model (A_MILP) outperforms the network flow-based model (NF_MILP). Indeed, within 1 h of CPU time, A_MILP model gives optimal solutions for all real-world and generated instances in less than 188.49 s in average, whereas NF_MILP model solves all real-world instances but only 56.7 % of the generated instances within 348.37 s in average.

Moreover, the experimental results of the multi-objective problem validate the proposed hierarchical approach that minimises the patients' average length of stay while setting the number of patients' overnight stays to its minimum value. Indeed, patients' average length of stay kept its minimum value for 92.5 % of the solved test-bed instances and has increased by less than 0.11 % for the remaining instances.

In practice, surgical process times are highly variable and uncertain. Thus, accounting for this random behaviour in patients' scheduling is crucial. For future research, we aim to develop a stochastic approach based on the proposed deterministic assignment model (A_MILP).

Furthermore, in day-to-day operation, surgical process times are often subject to disruptions that perturb the optimized schedules and causes delays and cancellations. Thus, an interesting perspective is to build robust schedules having less likelihood to be damaged by unpredicted events. This is a topic of our ongoing research.

Acknowledgments This research has been conducted within a MOBIDOC thesis framework of PASRI program that is supported in part by the *European Union* and *Clinique Ennasr*.

References

1. Hopkins L, Labonté R, Runnels V, Packer C (2010) Medical tourism today: what is the state of existing knowledge? J Public Health Policy 31(2):185–98. doi:10.1057/jphp.2010.10
2. O'Neill L, Dexter F, Wachtel RE (2009) Should anesthesia groups advocate funding of clinics and scheduling systems to increase operating room workload? Anesthesiology 111:1016–1024
3. Pham D-N, Klinkert A (2008) Surgical case scheduling as a generalized job shop scheduling problem. Eur J Oper Res 185:1011–1025. doi:10.1016/j.ejor.2006.03.059
4. Xiang W, Yin J, Lim G (2015) A short-term operating room surgery scheduling problem integrating multiple nurses roster constraints. Artif Intell Med 63:91–106. doi:10.1016/j.artmed.2014.12.005
5. Fei H, Meskens N, Chu C (2010) A planning and scheduling problem for an operating theatre using an open scheduling strategy. Comput Ind Eng 58:221–230. doi:10.1016/j.cie.2009.02.012
6. Guinet A, Chaabane S (2003) Operating theatre planning. Int J Prod Econ 85:69–81. doi:10.1016/S0925-5273(03)00087-2
7. Fei H, Meskens N, Chu C (2006) An operating theatre planning and scheduling problem in the case of a block scheduling strategy. Int Conf Serv Syst Serv Manag. doi:10.1109/ICSSSM.2006.320500
8. Jebali A, HadjAlouane AB, Ladet P (2006) Operating rooms scheduling. Int J Prod Econ 99:52–62. doi:10.1016/j.ijpe.2004.12.006
9. Roland B, Di Martinelli C, Riane F, Pochet Y (2010) Scheduling an operating theatre under human resource constraints. Comput Ind Eng 58:212–220. doi:10.1016/j.cie.2009.01.005
10. Liu Y, Chu C, Wang K (2011) A new heuristic algorithm for the operating room scheduling problem. Comput Ind Eng 61:865–871. doi:10.1016/j.cie.2011.05.020
11. Marques I, EugéniaCaptivo M, VazPato M (2012) An integer programming approach to Elective surgery scheduling analysis and comparison based on a real case. OR Spectr 34:407–427. doi:10.1007/s00291-011-0279-7
12. Marques I, EugéniaCaptivo M, VazPato M (2014) Scheduling elective surgeries in a Portuguese hospital using a genetic heuristic. Oper Res Health Care 3:59–72. doi:10.1016/j.orhc.2013.12.001
13. Cardoen B, Demeulemeester E, Beliën J (2010) Operating room planning and scheduling: a literature review. Eur J Oper Res 201: 921–932. doi:10.1016/j.ejor.2009.04.011
14. Francesca G, Rosita G (2011) Operational research in the management of the operating theatre: a survey. Health Care Manag Sci 14: 89–114. doi:10.1007/s10729-010-9143-6
15. Pariente JMM, Torres JMF, Cia TG (2009) Policies and decision models for solving elective case Operating Room scheduling. Comput Ind Eng 112 – 117. doi:10.1109/ICCIE.2009.5223538
16. Guo M, Wu S, Li B, Song J, Ron Y (2014) Integrated scheduling of elective surgeries and surgical nurses for operating room suites. Flex Serv Manuf J 1–16. doi:10.1007/s10696-014-9199-6
17. Beliäen J, Demeulemeester E (2008) A branch-and-price approach for integrating nurse and surgery scheduling. Eur J Oper Res 3:652–668. doi:10.1016/j.ejor.2006.10.060
18. Cardoen B, Demeulemeestera E, Belien J (2009) Sequencing surgical cases in a day-care environment: an exact branch-and-price approach. Comput Oper Res 36:2660–266. doi:10.1016/j.cor.2008.11.012
19. Meskens N, Duvivier D, Hanset A (2012) Multi-objective operating room scheduling considering desiderata of the surgical team. Decis Support Syst 55:650–659. doi:10.1016/j.dss.2012.10.019

20. Wang T, Meskens N, Duvivier D (2015) Scheduling operating theatres: mixed integer programming vs Constraint programming. Eur J Oper Res 247:401–413. doi:10.1016/j.ejor.2015.06.008
21. Augusto V, Xie X, Perdomo V (2010) Operating theatre scheduling with limited recovery beds and patient recovery in operating rooms. Comput Ind Eng 58:231–238. doi:10.1016/j.cie.2009.04.019
22. Zhao Z, Li X (2014) Scheduling elective surgeries with sequence-dependent setup times to multiple operating rooms using constraint programming. Oper Res Health Care 3:160–167. doi:10.1016/j.orhc.2014.05.003
23. Min D, Yih Y (2010) Scheduling elective surgery under uncertainty and downstream capacity constraints. Eur J Oper Res 206:642–652. doi:10.1016/j.ejor.2010.03.014
24. Wang Y, Tang J, Richard YK, Fung A (2014) Column-generation-based heuristic algorithm for solving operating theatre planning problem under stochastic demand and surgery cancellation risk. Int J Prod Econ 158:28–36. doi:10.1016/j.ijpe.2014.07.015
25. Hans E, Wullink G, Van Houdenhoven M, Kazemier G (2008) Robust surgery loading. Eur J Oper Res 185:1038–1050. doi:10.1016/j.ejor.2006.08.022
26. Patrick J, Puterman ML, Queyranne M (2008) Dynamic multipriority patient scheduling for a diagnostic resource. Oper Res 56(6):1507–1525. doi:10.1287/opre.1080.0590
27. Persson M, Persson JA (2009) Health economic modeling to support surgery management at a Swedish hospital. Omega 37:853–863. doi:10.1016/j.omega.2008.05.007
28. Patrick J, Puterman ML (2007) Improving resource utilization for diagnostic services through flexible inpatient scheduling: A method for improving resource utilization. J Oper Res Soc 58:235–245. doi:10.1057/palgrave.jors.2602242
29. López J.A, López C.I, Olguín J.E et al. (2013) Surgery Scheduling Using Simulation with Arena. World Acad Sci Engi Technol Vol:7 2013-07-24
30. Prudtikul P, Pathomsiri S (2013) Study and Design of Patient Flow at the Medicine Department of a University Hospital. World Acad Sci Eng Technol Vol:7 2013-02-22
31. Saremi A, Payman J et al (2013) Appointment scheduling of outpatient surgical services in a multistage operating room department. Int J Prod Econ 141(2):646–658. doi:10.1016/j.ijpe.2012.10.004
32. Wang T, Guinet A and Besombes B (2009) A sizing tool for allocation planning of hospital bed resources. Intell Patient Manag 113–125. doi: 10.1007/978-3-642-00179-6_7
33. Thompson S, Nunez M et al (2009) OR practice-efficient short-term allocation and reallocation of patients to floors of a hospital during demand surges. Oper Res 57(2):261–273. doi:10.1287/opre.1080.0584
34. Ben Bachouch R, Guinet A, Hajri-Gabouj S (2012) An integer linear model for hospital bed planning. Int J Prod Econ 140(2): 833–843. doi:10.1016/j.ijpe.2012.07.023
35. Berge Holm L, 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. doi:10.1016/j.ijmedinf.2012.05.006
36. Gupta JND, Tunic EA (1991) Schedules for a two-stage hybrid flowshop with parallel machines at the second stage. Int J Prod Res 29:1489–1502. doi:10.1023/A:1018976827443
37. Ruiz R, Vázquez-Rodríguez JA (2010) The hybrid flow shop scheduling problem. Eur J Oper Res 205(1):1–18. doi:10.1016/j.ejor.2009.09.024
38. Bertel S, Billaut JC (2004) A genetic algorithm for an industrial multiprocessor flow shop scheduling problem with recirculation. Eur J Oper Res 159:651–662. doi:10.1016/S0377-2217(03)00434-X
39. Boudhar M, Meziani N (2010) Two-stage hybrid flow shop with recirculation. Int Trans Oper Res 17:239–255. doi:10.1111/j.1475-3995.2009.00711.x
40. Lin H-T, Liao C-J (2003) A case study in a two-stage hybrid flow shop with setup time and dedicated machines. Int J Prod Econ 86(2):133–143. doi:10.1016/S0925-5273(03)00011-2
41. Yang J (2010) A new complexity proof for the two-stage hybrid flow shop scheduling problem with dedicated machines. Int J Prod Res 48(5):1531–1538. doi:10.1080/00207540802691374
42. Wachtel RE, Dexter F (2009) Influence of the operating room schedule on tardiness from scheduled start times. Anesth Analg 108:1889–1901
43. Sulecki L, Dexter F, Zura A, Saager L, Epstein RH (2012) Lack of value of scheduling processes to move cases from a heavily used main campus to other facilities within a healthcare system. Anesth Analg 115:395–401
44. Sherali HD, Adams WP (1990) A hierarchy of relaxations between the continuous and convex hull representations for zeroone programming problems. SIAM J Discrete Math 3(3):411–430
45. Sherali HD, Adams WP (1994) A hierarchy of relaxations and convex hull characterizations for mixed-integer zero–one programming problems. Discrete Appl Math 52(1):83–106
46. http://www.santetunisie.rns.tn/fr/cahier_charge/c_charge_fr/C20_fr.pdf . Accessed 29 Jan 2016
47. http://new.sfar.org/article/13/recommandations-concernant-la-surveillance-post-anesthesique . Accessed 29 Jan 2016
48. http://support.minitab.com/en-us/minitab/17. Accessed 16 Aug 2016
49. Zhou J, Dexter F (1998) Method to assist in the scheduling of add-on surgical case: upper prediction bounds for surgical case durations based on the log-normal distribution. Anesthesiology 89(5):1228–1232
50. Spangler W, Strum D, Vargas L, May J (2004) Estimating procedure times for surgeries by determining location parameters for the lognormal model. Health Care Manag Sci 7(2):97–104