Noname manuscript No.
(will be inserted by the editor)

Dynamic Patient Admission Scheduling with Operating Room Constraints, Flexible Horizons, and Patient Delays

Sara Ceschia · Andrea Schaerf

Received: date / Accepted: date

Abstract We revisit and extend the Patient Admission Scheduling problem, in order to make it suitable for practical applications. The main novelty is that we consider constraints on the utilisation of operating rooms for patients requiring a surgery. In addition, we propose a more elaborate model that includes a flexible planning horizon, a complex notion of patient delay, and new components of the objective function.

We design a solution approach based on local search, which explores the search space using a composite neighbourhood. In addition, we develop an instance generator that uses real-world data and statistical distributions so as to synthesise realistic and challenging case studies, which are made available on the web along with our solutions and the validator.

Finally, we perform an extensive experimental evaluation of our solution method including statistically-principled parameter tuning, and an analysis of some features of the model and their corresponding impact on the objective function.

Keywords Patient Admission Scheduling · Patient Bed Assignment · Operating Rooms · Local Search

1 Introduction

The *Patient Admission Scheduling* (PAS) problem consists of scheduling patients within a planning horizon into hospital rooms so as to maximise management efficiency and patient comfort, and contribute to improve medical treatment effectiveness.

S. Ceschia · A. Schaerf
DIEGM, University of Udine
via delle Scienze 206, I-33100, Udine, Italy
E-mail: {sara.ceschia,schaerf}@uniud.it

The PAS problem is an extension of the *Patient Bed Assignment* (PBA) problem formalised by Demeester et al (2010) and further studied by Bilgin et al (2012). In previous work (Ceschia and Schaerf, 2011), we solved the PBA problem using local search and obtained the best-known solutions on all of the available PBA benchmarks (Demeester, 2009). Some of these results (on small problem instances) were successively improved by Range et al (2014).

Dynamic versions of the PBA problem have been studied by Ceschia and Schaerf (2012) and Vancroonenburg et al (2012). These versions not only introduced the dynamic registration of new patients, but also refined and extended the problem formulation to take into account several additional features, such as the presence of urgent patients and the uncertainty in stay lengths.

In our previous work cited above, we also considered the possibility of delaying patient admission, which implies the necessity to decide, besides the room for the hospital stay, the admission date of the patient. Thus, the problem falls outside the scope of the PBA problem and within the scope of the PAS problem.

In order to make the solutions of the PAS problem applicable in real-world situations, it is also necessary to consider the use of the operating theatre, one of the scarcest resources in a hospital.

As pointed out by Blake and Carter (1997) and Cardoen et al (2010), operating rooms are a critical resource, so the scheduling of surgeries has a deep impact on many other hospital activities. Indeed, a patient that undergoes surgery is expected to recover in the hospital, thus requiring a bed in a ward of the corresponding medical specialty. As a consequence, surgical scheduling must be integrated with patient admission in order to avoid resource conflicts.

To this aim, we present a further refinement of the PAS problem that includes, among other features, constraints on the utilisation of operating rooms and a new model for managing patient delays. This new problem formulation constitutes a significant step towards the full integration of patient admission scheduling with the surgery scheduling process, which is needed for a full-fledged application to be effectively used in practice.

We have designed a search method based on local search that solves this problem, along the lines of the one developed in our previous work (Ceschia and Schaerf, 2012). The new features, however, call for the definition of different neighbourhoods in order to solve large problem instances.

Given the difficulty in obtaining real data about patients (due to privacy issues), in order to test our search method we have designed an instance generator that uses real-world data about specialties and treatments, and statistical distributions about patients and operations. The generator, properly tuned, is able to create realistic and challenging cases for a large range of sizes for the main features of the problem. The generator, the generated instances, along with our solutions and a solution validator, are available on the web at the address `http://bitbucket.org/satt/or-pas` for future comparisons.

We have performed an experimental analysis to tune the solver so as to identify the best configuration of the parameters with sufficient statistical confidence. In addition, we have compared the results of the dynamic solver with those of the static one, for which all the events are known in advance. The outcome is that the difference in the performances is relatively small. Finally, we have performed some additional experiments to illustrate how the behaviour of the solver depends on the length of the planning horizon.

2 Problem formulation

The complete problem formulation is rather complex, therefore we introduce it in stages. First, we present the basic patient admission scheduling problem (Section 2.1), then we introduce the constraints related to the operating rooms (Section 2.2), and afterwards we describe how to model the initial conditions at the hospital and the planning horizon (Sections 2.3 and 2.4). In Section 2.5, we discuss the dynamic version of the problem, and finally in Section 2.6 we summarise the complete formulation.

2.1 Basic problem

The main entities involved in the patient admission scheduling problem are the following:

Patient: Each patient has an *expected* and a *maximum* admission date, and a length of stay (LoS) within the planning horizon. In addition, a patient has an *overstay risk* depending on the condition of her health, which represents the possibility of needing to spend extra nights in the hospital.

Day: The granularity of time for expressing the length of planned stays is a day; the set of (consecutive) days included in the problem is the *planning horizon*.

Room: The number of beds in a room is its *capacity* (typically one, two, or four). Each room belongs to a unique department. Each room has a different equipment (oxygen, telemetry, TV, ...). Patients may *need* or simply *desire* specific room equipment.

Treatment/Specialty: Each patient needs one treatment that belongs to a specific specialty (or specialism). Each department is qualified for the treatment of diseases of various specialties, but at different levels of expertise. That is, each department/specialty pair maps to one value in the set {Complete, Partial, None}. If the value is None, the department cannot host a patient that needs a treatment belonging to the given specialty; if it is Partial, she can be hosted, but with a penalty; if it is Complete, the patient can be hosted with no penalty.

In addition, the hospital has a *gender policy*. There are four different types of rooms, identified by the letters in the set {D, F, M, N}. In rooms of type F (resp. M) only female (resp. male) patients can be accepted. If the type is D the room can be occupied by patients of either gender, but on any given day the patients in the room must all be of the same gender. Finally, rooms of type N can be occupied simultaneously by patients of either gender (e.g., intensive care rooms).

Furthermore, some departments (paediatrics and geriatrics) are specialised in treating patients of a specific age range. For these departments, there is a limit on the minimum or the maximum age of the patients admitted.

Based on the above-mentioned features, we construct the integer-valued *Patient-Room matrix*, which represents the penalty of the assignment of the patient to the room. By convention, the value -1 means that the assignment is impossible because the room is not suitable, a positive value c means that the room is suitable but with a cost c, whereas 0 corresponds to a perfect fit in the room.

This matrix allows us to group all the constraints (hard and soft) related to departments, specialties, room equipment, age policy, room policy (types F and M only), and patient preferences. Indeed, all these constraints, with their weights, are included in this matrix of penalties for assigning a given patient to a given room.

The problem thus consists in assigning to each patient a room and an actual admission date, which can be later (but not earlier) than the expected one. The solution has to satisfy the following hard constraints:

Room Capacity (RC): The number of patients in a room cannot exceed its capacity.

Patient-Room Suitability (PRS): A patient cannot be assigned to a room that is not suitable (value -1 in the Patient-Room matrix).

The cost of a solution is given by the violations of the following soft constraints, together with others that will be introduced in the next sections.

Patient-Room Cost (PRC): If the value in the Patient-Room matrix is greater than 0, the assignment of a patient to the corresponding room generates a cost which is multiplied by the LoS of the patient and added to the objective function.

Room Gender (RG): For each day that a room of type D is occupied by males and females simultaneously, there is a penalty proportional to the size of the smaller of the two populations. Costs for rooms of types F and M are already included in the Patient-Room matrix. Rooms of type N generate no costs.

Delay (De): The delay of a patient admission generates a cost proportional to the length of the delay. In addition, given that a delay is more unpleasant if the expected admission is closer, the cost of the delay is multiplied by a *priority* which is inversely proportional to the closeness of the admission day.

Overcrowding Risk (Ri): A penalty is added in the case that a patient has an overstay risk and her room is fully occupied in the day after her discharge.

2.2 Operating room constraints

Operating rooms are assigned to specialties according to the so-called *Master Surgical Schedule* (MSS) (see, e.g., Beliën and Demeulemeester, 2007; Oostrum et al, 2009; Vissers et al, 2001). This is a cyclic timetable that defines, for each day of the cycle (typically a week), which operating room is assigned to a specialty and for how long.

In order to integrate MSS into the PAS problem, we introduce the notions of *operating room slot* and *surgery treatment*:

Operating Room Slot: An operating room (OR) slot is the smallest time unit for which an operating room can be reserved for a specialty in a day. Each day of the planning cycle, the MSS assigns to a specialty an integer number of OR slots. Different operating rooms can be simultaneously used by the same specialty, since a specialty normally does not identify a particular surgeon but rather a surgical group.

Surgery Treatment: Each patient has to undergo a medical treatment that is performed by a specialist. Some treatments include a surgery of the type of the corresponding specialty. In this case, the day of the surgery (typically either the day after the admission day or the admission day itself) and the expected length of the surgery must be specified, along with the specialty.

The assignment is thus subject to all the constraints defined in Section 2.1 and the following additional one:

Operating Room Utilisation (ORU): For each day and each specialty, the total length of the surgeries belonging to the specialty must not exceed the time granted to it by the MSS.

This condition is a hard constraint, and it has a critical impact on the search space. Consequently, it requires that we reconsider the search method, and in particular the definition of neighbourhood structures, with respect to previous work.

Note that we select only the admission day of a patient; the problem of sequencing OR slots in different operating rooms and surgeries within each OR slot is outside the scope of this work, and is usually performed online (at most the day before).

Special attention is required for the management of *urgent* patients. According to Wullink et al (2007) and Hans and Vanberkel (2012), urgent patients are operated in any available room, regardless of the specialty. To accommodate this policy, the length of surgeries for urgent patients are not included in the computation of the utilisation for the ORU constraint. On the other hand, the total occupation (summing all specialties, including urgencies) must be below the total capacity. For this reason, we include another constraint that deals specifically with this issue:

Operating Room Total Utilisation (ORTU): For each day, the total length of surgeries belonging to all specialties, including the urgent ones, must not exceed the total capacity of the operating rooms.

An example utilisation of operating room slots, and the corresponding violations of the ORU and ORTU

Table 1 An example of an MSS (OR time available) and actual utilisation, divided into elective cases (E) and urgent cases (U). Values are expressed in minutes.

	Monday	**Tuesday**	**Wednesday**
Vascular Surgery	MSS = 180 E = 210 U = 0	MSS = 180 E = 120 U = 60	MSS = 360 E = 300 U = 90
General Surgery	MSS = 360 E = 240 U = 60	MSS = 180 E = 300 U = 60	MSS = 360 E = 360 U = 30

constraints are shown in Table 1 and Figures 1 and 2.

Table 1 reports for three days and for two specialties, Vascular Surgery (VS) and General Surgery (GS), the available OR time assigned by the MSS, and the actual OR time spent to operate elective cases (E) and urgent cases (U).

Fig. 1 A representation of the example of Table 1 with the resulting ORU violations.

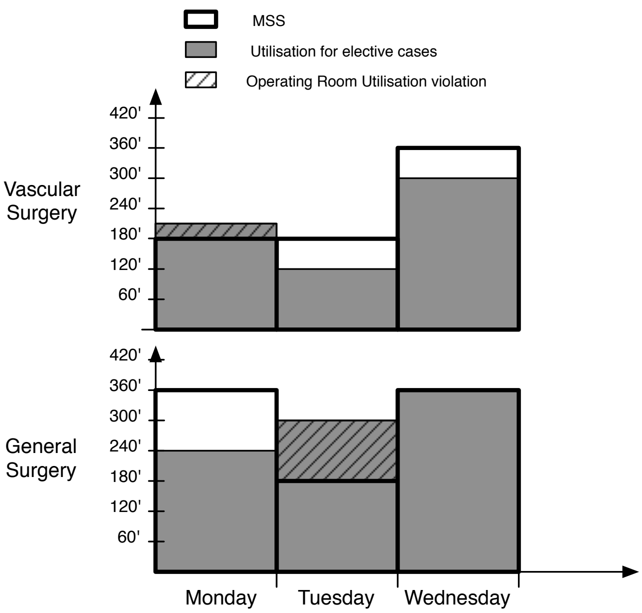
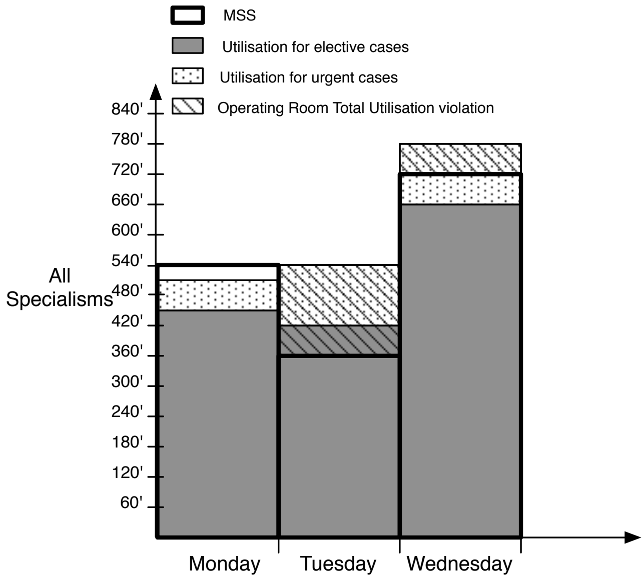


Figure 1 shows the violations of the ORU constraint for the situation described by the data in Table 1.

Notice that urgent patients do not contribute to the computation of the total utilisation of the specialty, therefore, for example, for VS we have on Monday a violation but not on Wednesday. In fact, on Monday the ORU violation is equal to 30 minutes $(210 - 180)$, whereas on Wednesday there is no ORU violation but 60 minutes of idle OR time $(360 - 300)$.

Figure 2 shows the violations of the ORTU constraint for the same situation. For each day, the available OR time is the sum of the times assigned by the MSS to each specialty, that is 540 minutes on Monday, 360 on Tuesday and 720 on Wednesday. For the ORTU constraint, all surgeries are counted, and therefore there is violation on Tuesday $(120+60+300+60 > 360)$ and Wednesday $(300 + 90 + 360 + 30 > 720)$. On Monday, however there is no violation, since the underused slots assigned to GS, compensate for the overuse of VS slots.

Fig. 2 A representation of the example of Table 1 with the resulting ORTU violations.



The ORU and ORTU constraints involve only the total length of slots. In reality, this length is divided into *normal time* and *overtime*. Normal time can be used freely, whereas overtime is allowed but should be minimised. To model this situation, the problem includes a cost component (soft) for the overtime work.

Operating Room Overtime (ORO): For each day and each specialty, the total length of the surgeries belonging to the specialty should not exceed the normal time granted to it.

When a specialty goes overtime, it is created a cost related to the dedicated personnel, but not necessarily for the general staff of the operating room, because some other specialty might not use its full time, thus balancing the occupancy.

To account for all costs, we include a second component that counts the total overtime of the rooms.

Operating Room Total Overtime (ORTO): For each day, the total length of surgeries belonging to all the specialties, including the urgent ones, should not exceed the total normal time of the operating rooms.

2.3 Initial situation

A hospital runs all year long. Therefore, for any scheduling period, it is necessary to take into account the patients that are already in the wards at the beginning of the period (called *initial patients*).

An initial patient needs to be scheduled like all other patients, with the following distinctive features:

- the admission day is equal to 0 (the initial planning day), and the LoS is equal to the residual period of the patient;
- the maximum delay is set to 0 (i.e., she cannot be delayed);
- the patient is already hosted in a specific room, and this fact needs to be taken into consideration.

If an initial patient is assigned by the new schedule to a room different than her previous one, a *transfer* is generated, which is a clear discomfort for the patient, and thus needs to be penalised by a new cost component (called Tr). Transfers are taken into account by adding to the Patient-Room matrix a new term for all rooms but the one previously occupied by the initial patient.

Notice that in this work, in contrast to Demeester et al (2010), we do not consider *planned transfers*, which are changes to a current patient's room assignment by the solver. Conversely, we do consider the possibility of changing the room of an initial patient (so-called *sudden transfers*). In fact, planned transfers are very rarely necessary, whereas the sudden ones are more frequent (due to emergencies).

2.4 Extended horizon

It is common in hospitals for the total requests to exceed the capacity of the resources in the planning horizon, at least for some specialties. In these cases, it is impossible to find a feasible solution. Therefore, it is usually necessary to delay some of the patient admissions even beyond the original planning horizon.

For this reason, we introduce the *extended horizon*, which allows the planner to delay patient admissions beyond the original horizon in order to obtain feasible solutions.

This practice however should be limited, not only because it creates discomfort for the patients, but also because it consumes resources from future planning periods. To this end, we introduce two new cost components that track the resources (rooms and OR slots) that are left idle in the original planning horizon. These components measure the difference of usage of the resource with respect to its maximum possible usage (which is the minimum between the total request and the total capacity).

Idle Rooms (IR) and **Idle OR Slots** (IOS): The difference between the actual usage of beds in rooms (resp. OR slots in operating rooms) and their maximum possible usage within the original planning horizon creates a cost, which is added to the objective function.

As observed experimentally in Section 6.4, in our setting the best length of the extended horizon is the double of the original one.

2.5 Dynamic problem

So far, we have described the *static* version of the problem, in which all information about admissions is known in advance. In practical situations, however, new events happen *dynamically*, requiring adjustments to the schedule (e.g. emergencies).

To model the dynamic situation, each patient in the input data is assigned a *registration day*, which is the day in which she becomes known to the system, and the expected and maximum admission dates are set.

In this setting, a patient is *urgent* whenever the registration day is the same as the admission day, and the maximum delay is 0; all others are called *elective*. In the instances we generated, we have an average of 10% of urgent patients, so as to adhere to a realistic load on the hospitals (Jebali et al, 2006a).

The problem is thus solved for every single day d in the original horizon. For each day d, before running the solver, the system adds the patients registered in that day. After running the solver, its result is stored, so that all discharged patients are removed and the patients admitted in day d are marked as initial patients. This iterative procedure is shown in Figure 3.

Notice that some (elective) patients are scheduled several times before they are actually admitted and their assignment becomes permanent. Therefore, the evaluation each day of the quality of the solution does not necessarily influence the final result. For this reason, the final cost can only be computed by evaluating the solution on the last iteration of the overall process.

2.6 Complete problem formulation

Collecting together all the cost components introduced in the previous sections, we have the complete formulation, in the static and dynamic versions. We do not

Fig. 3 Dynamic solution procedure (h: original horizon).

```
procedure DynamicSolve(h)
    d ← 0
    while d < h do
            // Insert patients registered at day d
        InsertPatients(d);
            // Solve the static problem (in day d)
        Solve(d);
            // Remove patients discharged in day d and
            // set all patients admitted in day d as initial
        UpdateSolution(d);
        d ← d + 1;
    end while
end procedure
```

report the full mathematical model here, which is an extension of the one shown in our previous work (Ceschia and Schaerf, 2012). However, a solution validator is available at our web site, so that there cannot be misunderstanding on the precise interpretation of the components.

The weights of the various components represent their nominal costs for the hospital; they are obviously crucial, as they influence the structure of the solutions, by giving priority to one or the other aspect of the problem. In practice, setting these weights would be the responsibility of hospital staff, based on professional experience. For experimental evaluation, however, we set them to the values reported in Table 2.

3 Related work

The literature on Operations Research applied to the health care domain is vast; it ranges from ambulatory and inpatient care services, to emergency units and operating theatre management, up to home care services, involving decisions at different hierarchical, temporal, and managerial levels (Hulshof et al, 2012).

In particular, a hierarchical division in strategic, tactical, and operational planning applies to most of health care services. Strategic planning addresses long-term decisions about design, sizing, and development of new facilities, and is usually based on historical data or forecasts. Tactical level manages the assignment of aggregated resources to patient groups or specialties. Finally, the operational level pertains to short-term decisions about the scheduling of single tasks, defining the individual patient and resources involved. When the decisions consider only the elective demand, it is called *offline* operational planning, whereas when emergencies (or any other unplanned events) are also taken into account resulting in a real-time (re)scheduling, it is called *online* operational planning.

Our model integrates, in a dynamic context, features of the patient admission scheduling problem with operating room scheduling, taking into account also uncertainty about the duration of the patient's hospitalisation and emergency cases. Indeed, health care systems are strongly affected by uncertainty, which typically characterises the duration of the surgeries, the recovery process and the arrivals of urgent cases (Cardoen et al, 2010; Guerriero and Guido, 2011).

Given that our work combines different problems, we present the literature review in two sections, corresponding to the main issues involved.

3.1 Patient admission problems

The patient bed assignment (PBA) problem, as first formulated by Demeester et al (2010), belongs to the field of inpatient care and it operates at the offline operational level, given that it deals with the assignment of beds to elective patients.

Due to the problem's complexity, it is usually tackled by approximate techniques: Demeester et al (2010) proposed a tabu search algorithm, Ceschia and Schaerf (2011) improved the results with a simulated annealing metaheuristic, Bilgin et al (2012) devised a hyperheuristic approach where the low-level components are local search moves; lastly, Range et al (2014) presented a new mathematical formulation and a column generation-based heuristic which allows them to compute tighter lower bounds and new best solutions for the smaller benchmarks.

The online version of PBA was introduced in (Ceschia and Schaerf, 2012), where the problem is dynamically solved by a metaheuristic method using daily rescheduling, considering also urgent patients and uncertainty about the LoS. That work also considers the possibility to delay the admission date of a patient.

Vancroonenburg et al (2012) also studied PBA in a dynamic context, introducing an estimated LoS for each patient and proposing two integer programming (IP) models: one considering only patient arrivals, and the other taking into account also patient registration without admission.

Schmidt et al (2013) developed a decision support system for bed management in hospital which considers patient priority, stochastic LoS and dynamic adjustment of the plan in response to uncertainty. The problem is formally described through an IP model and an exact approach is compared with various heuristic strategies evaluated in a simulation environment.

Gartner and Kolisch (2014) presented two mixed-integer programming (MIP) models for planning the

Table 2 Default values of the weights of the cost components.

Cost component	Accounting	Value
Missing room equipment (PRC1)	per day, per patient	20
Unsatisfied room preference (PRC2)	per day, per patient	10
Partial specialty level (PRC3)	per day, per patient	20
Unsatisfied gender policy (PRC4)	per day, per (misplaced) patient	50
Transfer (Tr)	per patient	100
Delay (De)	per day, per patient, per priority	5
Overcrowd Risk (Ri)	per patient	1
Idle Operating Room Slots (IOS)	per minute	10
Idle Room Capacity (IR)	per day, per bed	20
Overtime (ORO and ORTO)	per minute	3

patient flow in hospitals at the operational level: the first model has fixed admission dates, whereas the second considers the possibility of varying the admissions within a time window. The aim is to maximise the contribution margin. The problem is tackled in both static and dynamic contexts using a rolling horizon approach.

Most of the literature addressing the PAS is at the tactical level: patients are usually grouped depending on the treatment and resources that they require, and then some rules are defined to establish for each group the number of elective patients to admit to the hospital, with the overall goal of maximising resource usage and patient throughput. At the operational level, the problem consists in defining the date (and possibly the time) of the admission of each individual patient. Indeed, circumstances could require postponing some planned admissions of elective patients in order to handle urgencies or prolonged stays.

At the tactical level, a comparison between different hospital admission systems has been done by Vissers et al (2007). Using simulation, the authors examined the impact of different policies (maximum resource usage, zero waiting time, coordinated booked admission, uncoordinated booked admission) considering multiple performance criteria (resource utilisation, average waiting times, elective patients cancelled, urgent patient rejected, overuse).

Hutzschenreuter et al (2008) presented an agent-based model for the selection of an optimal mix for patient admissions. Nunes et al (2009) and Patrick et al (2008) modelled the admissions control problem as an infinite-horizon Markov Decision Process, the former in a static and the latter in a dynamic context. Min and Yih (2010a) developed a dynamic programming model for optimally selecting patients from the waiting list: Patients are selected according to different priorities and limits on operating room capacity. The objective function considers the trade-off between the cost of delaying the admission of a patient and the cost related to overtime, which is occurred when the total surgery duration exceeds the operating room capacity. The model also includes uncertainty in arrivals and surgery duration. Hulshof et al (2013) presented a MIP model embedded in a rolling horizon system to build an admission plan for different patient groups in multiple periods. They studied different performance metrics and proposed an iterative procedure, based on the achievement of targets set by hospital managers, to determine the weights of the components of the objective function.

3.2 Operating room assignment

The scheduling of operating rooms can also be decomposed into the three classic hierarchical stages (Guerriero and Guido, 2011). The aim of the strategic planning is to assign the operating room resources to surgical specialties according to historical data or forecasts. The tactical medium-term planning involves the construction of the master surgical schedule (MSS). Once the MSS has been set, the elective surgeries are planned and the corresponding elective patients are scheduled (offline operational planning). This process generates the demand of beds for the downstream wards, which will host the patients during recovery. The last step is the daily scheduling of the specific intervention for each patient. Emergencies and all kinds of uncertainty are tackled at this level, entailing the online operational planning. The operational level can be further divided in two main stages: the *advanced scheduling* that assigns patients to operating room slots, and the *allocation scheduling* that sequences the surgeries in the OR slot.

Our problem can be positioned at the advanced scheduling stage of the operational level, in that inputs to the problem are all of the data about departments, rooms and patients, as well as the MSS resulting from the tactical planning, while the output is the assignment of patients to rooms and OR slots, as well as the admission day. Using the classification of Jebali et al (2006b), the problem belongs to the class of “block scheduling", because the operating room scheduling is organised in blocks assigned to specialties.

Vancroonenburg et al (2013) developed independently a formulation similar to the one proposed here for scheduling patient admissions constrained by rooms and operating theatre capacities. In their formulation, however, only elective patients are considered, the context is static, and other cost components are modelled differently.

The offline advanced scheduling of patients in the operating theatre has been the object of various studies. Hans et al (2008) assigned additional planned slack to planned surgeries to obtain a surgery schedule more robust against overtime. The approach uses some constructive heuristics (list scheduling and sampling procedures based algorithms) and local search methods. Fei et al (2008) formulated the *Surgical Case Assignment* problem as a set partitioning problem and devised a branch-and-price algorithm to solve it exactly. Jebali et al (2006b) and Fei et al (2010) addressed the integration of advanced and allocation scheduling activities in a two-step decomposition approach: Jebali et al (2006b) modelled both problems as MIP problems; Fei et al (2010) used a column-generation-based heuristic procedure for the advanced scheduling, subsequently they applied a hybrid genetic algorithm to obtain the detailed daily sequencing. In contrast, Riise and Burke (2011) solved the combined problem by integrating both activities in one stage, using an iterated local search and variable neighbourhood descent.

The scheduling of urgent patients in operating rooms has been studied by Wullink et al (2007) and Hans and Vanberkel (2012). Both works came to the conclusion that sharing ORs with elective cases (instead of dedicated emergency ORs) leads to a substantial reduction in waiting times and overtime. To our knowledge, only Lamiri et al (2009) have solved cases where operating rooms are shared between elective and emergency surgeries. The authors formulated the planning problem as a stochastic IP model, and they solved it using MIP combined with Monte Carlo simulation. However, in the paper they claim that this solution approach is impracticable for large instances, thus different approximate methods (heuristics and metaheuristics) are also proposed and compared.

The impact of the MSS on the downstream activities, in particular on bed occupancy, has been previously addressed by different authors. Beliën and Demeulemeester (2007) and Beliën et al (2008) investigated various MIP models for building the MSS with the objective of obtaining a levelled bed occupancy, and compared them with some metaheuristic algorithms. Chow et al (2011) proposed a framework consisting of a simulation model that predicts the bed occupancy for a given MSS, and a MIP model that deals with the creation of the MSS.

Lastly, Vijayakumar et al (2013) devised a MIP model for the surgical scheduling problem which considers multiple periods, multiple resources (operating rooms, surgeons, nurses, equipment) and patients with various priority. Exploiting the analogy with the dual bin packing problem, the authors proposed also a first-fit decreasing algorithm able to solve larger real cases.

4 Solution technique

We design a solution technique based on local search. Therefore, we introduce all the key ingredients of a local search method: the search space, the cost function, the initial solution (Section 4.1), the neighbourhood relations (Section 4.2), and finally the metaheuristic technique used, namely simulated annealing (Section 4.3).

4.1 Search space, cost function, and initial solution

A state in the search space is defined by two integer-valued vectors: the first represents the room assigned to each patient, and the second is the scheduled delay of the patient's admission.

The cost function obviously includes all the (soft) cost components. In addition, violations of some hard constraint are also accepted. Therefore, as customary, we add a component called *distance to feasibility* that takes into account the violation of the room capacity (RC) and the operating room utilisation constraints (ORU and ORTU). These violations are inserted in the cost function with a suitably high weight. On the other hand, patient room suitability constraints (PRS) are never violated in our search space.

The initial solution simply assigns a random (but suitable) room to every patient, and delays are all set to zero.

4.2 Neighbourhood relations

In order to effectively explore the search space in which the admission of patients must satisfy the operating room constraint, we need to define *ad hoc* neighbourhoods. Indeed, we should consider the case where a specialty does not appear in the MSS every day. The neighbourhood relation defined in (Ceschia and Schaerf, 2012), which moves the admission of a patient only to the previous or following day, is not effective. We thus have to define a neighbourhood that enables longer time leaps in a single move.

Based on these considerations, the neighbourhood used is the composition of four basic move types. The first two operate on the room assignment. One changes the room assigned to a patient and the other swaps assigned rooms between two patients. The third and fourth are used to modify the admission day of patients. One shifts the admission of a patient forward or backward by a given number of days. The other swaps the admission day and the room between two patients.

ChangeRoom (CR): One patient is moved to a new room that is suitable for her.

SwapRooms (SR): Two patients exchange their rooms. Both rooms must be suitable for the patients involved.

ShiftAdmission (ShA): One patient admission is moved forward or backward in time by a fixed number of days. The admission can be neither before the planned one nor after the maximal delay. The room is left unchanged.

SwapAdmissions (SwA): The admission dates and the rooms of two patients are exchanged. Neither new admission can be before the planned one or after the maximum one for the newly assigned patient. The new rooms must be suitable for the patients.

For the SR neighbourhood, in order to reduce its size and thus focus on useful moves, we consider only pairs of patients such that in the current state there is at least one day in which they are both scheduled to be in the hospital (i.e. their planned stay overlap).

4.3 Metaheuristic

We use simulated annealing (SA) (Kirkpatrick et al, 1983) as it has been successfully used in the PBA problem before. For a detailed description of SA we refer to the work of van Laarhoven and Aarts (1987). Here we report the basic ingredients of our implementation.

Initial solution: The initial solution is randomly generated, as explained in Section 4.1, and as prescribed by SA.

Move selection: The move is selected at random, as customary in SA. However, given that we use a composite neighbourhood, the random selection is a two-stage process: first the basic neighbourhood is selected, then the specific move within the neighbourhood is chosen. The latter drawing is done in a uniform way, while the former is driven by four given probability values $p_{\mathtt{CR}}$, $p_{\mathtt{SR}}$, $p_{\mathtt{ShA}}$, and $p_{\mathtt{SwA}}$ (with sum 1). The values of the p_* probabilities are selected by experimental tuning, along with the other SA parameters.

Cooling scheme: The cooling scheme is the classic geometric one, with slope α and start temperature T_0 (both to be set experimentally). The temperature T is decreased when a fixed number of iterations N have been performed.

Acceptance criterion: Improving and sideways moves are always accepted, worsening ones are accepted with probability $e^{-\Delta f/T}$, where f is the cost function.

Stop criterion: The procedure stops when the system reaches a fixed final temperature T_{min}.

The control parameters of the procedure are the cooling rate (α), the number of neighbours sampled at each temperature (N), the starting and final temperatures (T_0 and T_{min}), and three out of four neighbourhood probability values (the last is computed).

5 Instance generator

Unfortunately, there are no real-world data sets available at present for this problem, therefore we develop an instance generator that can synthesise problem instances of any size and length.

5.1 Data and distributions

The generator is based on real-world data and distributions taken from the literature. The structural organisation in departments, the grouping of specialties, and the treatments are taken from real-world facilities. The mix of patients and the total number of OR slots available for each specialty is computed considering a distribution, which is randomly generated, that describes the relative importance of each specialty. The MSS is generated by a greedy algorithm that randomly selects a slot of a specialty and assigns it to the emptiest day of the period.

Both duration of surgery and LoS in the hospital are extracted from lognormal distributions (see Marazzi et al, 1998; Harper and Shahani, 2002; Min and Yih, 2010b) for each patient, while the arrivals follow a Poisson distribution with an expected value equal to the total number of patients divided by the number of days. Other features, such as age, gender, urgency, and treatment, are generated according to probabilities that are stored in a configuration file, which can be easily modified by the user. Scalar values that define the size of the generated instance, such as number of patients, number of rooms, and horizon, are set by command-line arguments.

Table 3 Main features of the generated families (5 instances per family).

Fam.	Rooms	Depts	OR	Specs	Treats	Patients	Days
Short1	25	2	2	9	15	391–439	14
Short2	50	4	4	18	25	574–644	14
Short3	75	6	5	23	35	821–925	14
Long1	25	2	2	9	15	693–762	28
Long2	50	4	4	18	25	1089–1169	28
Long3	75	6	5	23	35	1488–1602	28

In order to create a realistic initial situation, instead of creating the corresponding values at random, we resort to a simulation. That is, we create each instance seven days longer than requested, then we let our dynamic solver run for seven daily iterations. At the end of this period, we store the current state of the hospital as the initial situation of the instance and we remove the patients that have already been discharged during the first 7 days. We have verified a posteriori that the average occupancy has stabilised after this warm-up period of 1 week.

5.2 Instances

We have generated 30 instances, divided into six families whose main characteristics are shown in Table 3. For patients we report minimum and maximum values, given that the number is not fixed. In fact, each day the generator samples the number of arrivals from a Poisson distribution and then the solver runs for the warm-up period; as a consequence we cannot set the total number of patient to a pre-determined value. The first three families are short cases (two weeks) of increasing size, and the following three are long cases (four weeks) following the same sizes of the infrastructure, but with a larger number of patients.

The main features are tuned experimentally so as to have an average occupancy in the original planning horizon of approximately 110%, in terms of both beds and operating rooms. This obviously implies that delays are necessary, which is however customary in today's hospitals.

The number of specialties, and consequently the number of treatments, increases with the number of departments. Indeed, each department hosts diverse specialties, that can be only medical (e.g. General Medicine), only surgical (e.g. Vascular Surgery), or both medical and surgical (e.g. Orthopaedics), depending on the type of treatments offered.

With the aim of computing the total number of OR slots available each day, we considered that each operating room is opened up to 9 hours in a day, divided in three slots of 180 minutes each.

Each instance is stored in a single XML file, which is validated against a corresponding XML schema. All instances and best solutions are available from the website `http://bitbucket.org/satt/or-pas`, along with the solution validator, the instance generator, and the XML schema.

6 Experimental analysis

The SA metaheuristic has several parameters, so a statistically-principled tuning phase is necessary in order to set the parameters to their most appropriate values for solving all instances.

In Section 6.1 we describe the tuning phase and in Section 6.2 we report our final results in terms of average and best scores. Then we move to additional experiments that highlight some specific features of the problem: In Section 6.3 we compare the results of our dynamic solver with those of the static one, and in Section 6.4 we test different lengths for the extended horizon.

The code is written in C++ and compiled using `gcc` v. 4.7.3, and all experiments have been performed on an Intel i7 CPU at 3.33GHz running Ubuntu 13.04.

6.1 Parameter tuning

As already mentioned, our composite-neighbourhood SA has several parameters to be tuned. This tuning is complicated by the fact that the running time could be very different from configuration to configuration. In order to overcome this challenge and equalise approximately the running times, we fix the total number of iterations I and compute the parameter N from the others so that the total I is always the same. Larger instances obviously need more iterations, therefore we decided to set $I = 0.5 \times 10^7$, $I = 10^7$ and $I = 1.5 \times 10^7$ for instances with 2, 4 and 6 departments, respectively. This results in an average running time of 5.4, 11.8, and 28.8 seconds respectively for each day of the dynamic solution loop of Figure 3, which lasts for 14 and 28 days for short and long instances, respectively.

Among the other parameters, a preliminary screening led us to focus on two of the parameters of SA (namely T_0 and T_{min}) on the one hand, and on the three independent probabilities (p_*) of the neighbourhood selection on the other.

For each of these two sets of parameters, we tested separately 33 different configurations extracted according to the nearly orthogonal Latin hypercubes (NOLH)

Table 6 Best results related to the Patient-Room Cost (PRC) broken up into cost components.

Family	PRC1 room equipment	PRC2 room preference	PRC3 specialty level	PRC4 gender policy
Short1	7993.60	3286.27	1186.93	99.00
Short2	15668.40	7434.00	5191.73	91.67
Short3	18770.40	11124.33	12886.40	39.00
Long1	22650.40	7959.53	1922.80	223.00
Long2	32531.60	16519.13	14710.00	198.00
Long3	49690.00	26611.27	43686.13	302.00

method proposed by Cioppa and Lucas (2007). We compared the results using the F-Race procedure (Birattari, 2004), which is based on the Friedman test to prove statistical significance, using the threshold of confidence set to 95% (p-value = 0.05).

The resulting configuration is shown in Table 4. All of the following experiments were performed using this configuration.

6.2 Best results

Table 5 shows the average costs for each family for the best configuration. Individual costs are posted on our website for each instance. The table also shows the contribution of each cost component. For the given setting of weights, the largest costs come from the unused resources (IR and IOS), and from the Patient-Room Cost (PRC). The cost of transfers (Tr) is also significant, along with the cost of delays (De). The other components are almost negligible.

For a better understanding of the Patient-Room Cost, in Table 6 we break this into its original components. This table shows that the main component comes from missing equipment of the rooms.

The fact that a large part of the cost comes from the component PRC is not a surprise, as the same happens for PBA as well, showing that a given mismatch between patient needs and room equipment is rather normal. Similarly, the large share coming from IR and IOS shows that, for the same reason, it is rather impossible to have a perfect match between patient mix and MSS, and thus to obtain a very high level of occupancy.

6.3 Static vs. dynamic solution

It is possible to define the *static* version of each instance, simply by changing the registration day of all patients to 0. This is a sort of "crystal ball" setting, in which every event is known in advance (including future urgent patients). With this setting, a single execution of the solver is sufficient to obtain the final solution.

Table 7 Comparison of static and dynamic average costs.

Family	Static	Dynamic	Δ %
Short1	82796.67	82490.51	-0.37
Short2	134655.97	135240.22	0.43
Short3	211859.40	214284.35	1.14
Long1	165622.99	169134.89	2.12
Long2	284873.29	292724.78	2.76
Long3	454616.24	468806.99	3.12

Comparing the solutions found by the static solver with those found by the procedure of Figure 3 sheds some light on the importance of having information in advance.

The iterations granted to the static solver are in line with those I granted to the dynamic solver, but proportional to the number of patients. That is, we multiply I by the number of patients in the static case, and we divide it by the average number of patients in the dynamic case. This way, the running time is approximately the same.

The average results are shown in Table 7, for the six families. The figures show that the gap increases with the size of the hospital and the length of the planning horizon. Not surprisingly, a larger increase of cost for the dynamic case is associated with the longer horizon.

The fact that the dynamic solver works better than the static one for the family `Short1` is due to the static solver not being allowed to transfer patients during the planning period, but only for initial patients.

Table 8 compares the costs of two specific components: the PRC and Tr alone. As expected, the number of transfers is much larger for the dynamic case, given that in the static case they can occur only in the first day of planning. In contrast, the other Patient-Room Costs are lower for the dynamic case. This shows that in general the discomfort of a transfer is balanced by the accommodation in a more suitable room.

As a general comment, the difference between the two cases (static vs. dynamic) is not considerably high, meaning that the margin for improvements to the dynamic solver by using some predictive technique for managing uncertainty is relatively small.

6.4 Extended horizon

In the previous experiments, the horizon has been extended to be the exact double of the original one. In Table 9 we show the results for various length of the extended horizon.

Table 4 Parameter setting emerged from F-Race.

Name	Role	fixed by	value
T_0	Start temperature	NOLH + F-Race	154.88
T_{min}	Final temperature	NOLH + F-Race	1.54
α	Cooling rate	preliminary experiments	0.999
N	Iterations per temperature	computed from I, T_0, T_{min}, and α	1086
$p_{\mathtt{CR}}$	Probability of CR moves	NOLH + F-Race	0.49
$p_{\mathtt{SR}}$	Probability of SR moves	NOLH + F-Race	0.35
$p_{\mathtt{ShA}}$	Probability of ShA moves	NOLH + F-Race	0.01
$p_{\mathtt{SwA}}$	Probability of SwA moves	computed from the other p_* values	0.15

Table 5 Best results broken up into cost components.

Family	PRC	RG	Tr	De	Ri	OT	IR	IOS	Total
Short1	12565.80	101.33	804.67	3456.20	71.35	1136.10	16541.07	47814.00	82490.51
Short2	28385.80	175.33	2748.67	4587.13	90.72	1309.10	34566.80	63376.67	135240.22
Short3	42820.13	326.33	4574.00	8054.93	140.85	2467.70	51613.07	104287.33	214284.35
Long1	32755.73	535.00	7114.67	10022.73	153.89	1555.40	34925.47	82072.00	169134.89
Long2	63958.73	897.33	17614.00	14457.20	209.32	3582.80	73918.40	118087.00	292724.79
Long3	120289.40	1760.33	30086.00	23926.87	285.13	3515.80	112393.47	175490.00	468806.99

Table 8 Comparison of static and dynamic average costs for the Patient-Room Cost (PRC) and Transfer (Tr).

Family	PRC			Tr		
	Static	Dynamic	$\Delta\%$	Static	Dynamic	$\Delta\%$
Short1	14173.53	12565.80	-11.34	238.67	804.67	237.15
Short2	31464.53	28385.80	-9.78	533.33	2748.67	415.38
Short3	47022.20	42820.13	-8.94	808.00	4574.00	466.09
Long1	37412.87	32755.73	-12.45	143.33	7114.67	4863.72
Long2	75375.73	63958.73	-15.15	659.33	17614.00	2571.49
Long3	135981.40	120289.40	-11.54	1215.33	30086.00	2375.53

Table 9 Comparison of results (median values) for different lengths of the extended horizon.

Family	Extended Horizon							
	+50%		+100%		+150%		+200%	
	Viol	Cost	Viol	Cost	Viol	Cost	Viol	Cost
Short1	510	87160	0	84726	0	84302	0	84398
Short2	470	135110	0	131573	0	131173	0	130684
Short3	1760	225298	0	211971	0	210372	0	210423
Long1	735	177571	0	175958	0	175308	0	173851
Long2	840	286959	0	283023	0	282332	0	281337
Long3	1785	490499	0	469874	0	462149	0	461621

The table highlights that for +50% (smaller than double horizon), the results are significantly worse, with several violations of the hard constraints. Conversely, for longer extended horizons the results remain, in most cases, basically constant. Actually, in some cases, they are even worse. This is due to the fact that the solver is heuristic, and might end up in a worse solution when given a larger space to explore. The optimal solution would obviously be non-increasing cost while moving to longer horizons, given that solutions that use the shorter horizon are feasible also for the longer one.

7 Conclusions and future work

The patient admission scheduling problem is a complex optimisation task that requires careful modelling in order to be applicable in practice. In addition, it is not a "stand-alone" problem, but rather is embedded in the life cycle of the hospital and tightly coupled with other tasks. Therefore, its model cannot be formulated independently from the rest of the activities at the hospital.

This work is a step towards the development of an optimisation-intensive industrial application in this field, that, to the best of our knowledge, does not yet exist. To this aim, we are working with an industrial partner (EasyStaff s.r.l.) in developing the infrastructure and user interface of the prospective application.

The outcome of this work is that operating rooms can be effectively incorporated in a model of patient admission, but they entail a more elaborate management of delays and more complex neighbourhood relations.

The development of the statistically-sound instance generator is also a significant part of the work, given that relatively few real-world benchmarks are available for this category of problems.

The experimental results are unfortunately incomparable with the literature, because they regard a newly-formulated problem and original instances. Nevertheless, a similar approach has proved to be effective for the simpler PBA problem (Ceschia and Schaerf, 2011), as it still holds the best results for large instances. In addition, all the instances and the results for this new problem are publicly available on the web, so that they are exposed to future comparison from the community.

For the future, we hope to be able to cooperate with other research groups working on the field to come up with a joint problem formulation that includes all the useful features, and could be used as a benchmark.

The next steps towards the definition of a more comprehensive and integrated problem formulation, could concern both upstream tasks, such as creating the MSS, and downstream activities, such as scheduling the individual surgeries in the operating room sessions (Testi and Tànfani, 2009; Guerriero and Guido, 2011).

In addition, together with our industrial partner we plan to conduct a set of interviews in hospitals so as to refine the important features and their relative weights.

Finally, we plan to develop a version of the problem in which the objectives related to the hospital management budget and those related to the patient comfort are separated and optimised in a multi-objective fashion. This process could shed some light on the trade-off between actual costs and level of service.

Acknowledgements We thank Paolo Sivilotti for proofreading the paper, and providing some useful comments.

Sara Ceschia acknowledges support from *Consorzio per l'AREA di ricerca scientifica e tecnologica di Trieste* and *Fondo Sociale Europeo in Friuli Venezia Giulia* under the program "S.H.A.R.M. - Supporting Human Assets in Research and Mobility".

References

Beliën J, Demeulemeester E (2007) Building cyclic master surgery schedules with leveled resulting bed occupancy. European Journal of Operational Research 176(2):1185–1204

Beliën J, Demeulemeester E, Cardoen B (2008) A decision support system for cyclic master surgery scheduling with multiple objectives. Journal of Scheduling 12(2):147–161

Bilgin B, Demeester P, Misir M, Vancroonenburg W, Vanden Berghe G (2012) One hyper-heuristic approach to two timetabling problems in health care. Journal of Heuristics 18(3):401–434

Birattari M (2004) The problem of tuning metaheuristics as seen from a machine learning perspective. PhD thesis, Université Libre de Bruxelles, Belgium

Blake JT, Carter MW (1997) Surgical process scheduling: a structured review. Journal of the Society for Health Systems 5(3):17–30

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

Ceschia S, Schaerf A (2011) Local search and lower bounds for the patient admission scheduling problem. Computers & Operations Research 38(10):1452–1463

Ceschia S, Schaerf A (2012) Modeling and solving the dynamic patient admission scheduling problem under uncertainty. Artificial Intelligence in Medicine 56(3):199–205

Chow VS, Puterman ML, Salehirad N, Huang W, Atkins D (2011) Reducing surgical ward congestion through improved surgical scheduling and uncapacitated simulation. Production and Operations Management 20(3):418–430

Cioppa TM, Lucas TW (2007) Efficient nearly orthogonal and space-filling latin hypercubes. Technometrics 49(1):45–55

Demeester P (2009) Patient admission scheduling website. `http://allserv.kahosl.be/~peter/pas/`, accessed 24 September 2013

Demeester P, Souffriau W, De Causmaecker P, Vanden Berghe G (2010) A hybrid tabu search algorithm for automatically assigning patients to beds. Artificial Intelligence in Medicine 48(1):61–70

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

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

Gartner D, Kolisch R (2014) Scheduling the hospital-wide flow of elective patients. European Journal of Operational Research 233(3):689–699

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

Hans EW, Vanberkel PT (2012) Operating theatre planning and scheduling. In: Handbook of Healthcare System Scheduling, vol 168, Springer, chap 5, pp 105–130

Hans EW, Wullink G, van Houdenhoven M, Kazemier G (2008) Robust surgery loading. European Journal of Operational Research 185(3):1038–1050

Harper PR, Shahani AK (2002) Modelling for the planning and management of bed capacities in hospitals. The Journal of the Operational Research Society 53(1):11–18

Hulshof PJ, Boucherie RJ, Hans EW, Hurink JL (2013) Tactical resource allocation and elective patient admission planning in care processes. Health Care Management Science 16(2):152–166

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

Hutzschenreuter AK, Bosman PAN, Blonk-Altena I, van Aarle J, La Poutré H (2008) Agent-based patient admission scheduling in hospitals. In: Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems: Industrial Track, International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, AAMAS '08, pp 45–52

Jebali A, Hadjalouane A, Ladet P (2006a) Operating rooms scheduling. International Journal of Production Economics 99(1-2):52–62, DOI 10.1016/j.ijpe.2004.12.006

Jebali A, Hadjalouane A, Ladet P (2006b) Operating rooms scheduling. International Journal of Production Economics 99(1-2):52–62

Kirkpatrick S, Gelatt CD Jr, Vecchi MP (1983) Optimization by simulated annealing. Science 220:671–680

van Laarhoven PJM, Aarts EHL (1987) Simulated annealing: theory and applications. Kluwer Academic Publishers

Lamiri M, Grimaud F, Xie X (2009) Optimization methods for a stochastic surgery planning problem. International Journal of Production Economics 120(2):400–410

Marazzi A, Paccaud F, Ruffieux C, Beguin C (1998) Fitting the distributions of length of stay by parametric models. Medical Care 36(6):915–27

Min D, Yih Y (2010a) An elective surgery scheduling problem considering patient priority. Computers & Operations Research 37(6):1091–1099

Min D, Yih Y (2010b) Scheduling elective surgery under uncertainty and downstream capacity constraints. European Journal of Operational Research 206(3):642–652

Nunes LGN, de Carvalho SV, de Cassia Meneses Rodrigues R (2009) Markov decision process applied to the control of hospital elective admissions. Artificial Intelligence in Medicine 47(2):159 – 171

Oostrum JM, Bredenhoff E, Hans EW (2009) Suitability and managerial implications of a master surgical scheduling approach. Annals of Operations Research 178(1):91–104

Patrick J, Puterman ML, Queyranne M (2008) Dynamic multipriority patient scheduling for a diagnostic resource. Operations Research 56(6):1507–1525

Range T, Lusby R, Larsen J (2014) A column generation approach for solving the patient admission scheduling problem. European Journal of Operational Research 235(1):252–264

Riise A, Burke EK (2011) Local search for the surgery admission planning. Journal of Heuristics 17:389–414

Schmidt R, Geisler S, Spreckelsen C (2013) Decision support for hospital bed management using adaptable individual length of stay estimations and shared resources. BMC Medical Informatics and Decision Making 13:3

Testi A, Tànfani E (2009) Tactical and operational decisions for operating room planning: Efficiency and welfare implications. Health Care Management Science 12(4):363–373

Vancroonenburg W, De Causmaecker P, Vanden Berghe G (2012) Patient-to-room assignment planning in a dynamic context. In: Proc. of the 9th Int. Conf. on the Practice and Theory of Automated Timetabling (PATAT-2012), pp 193–208

Vancroonenburg W, De Causmaecker P, Spieksma F, Vanden Berghe G (2013) Scheduling elective patient admissions considering room assignment and operating theatre capacity constraints. In: Proceedings of the 5th International Conference on Applied Operational Research, Lecture Notes in Management Science, vol 5, pp 153–158

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

Vissers J, Bertrand J, De Vries G (2001) A framework for production control in health care organizations. Production Planning & Control 12(6):591–604

Vissers JM, Adan IJ, Dellaert NP (2007) Developing a platform for comparison of hospital admission systems: An illustration. European Journal of Operational Research 180(3):1290–1301

Wullink G, Houdenhoven M, Hans EW, Oostrum JM, Lans M, Kazemier G (2007) Closing emergency operating rooms improves efficiency. Journal of Medical Systems 31(6):543–546