Computers & Industrial Engineering 97 (2016) 248–257

Contents lists available at ScienceDirect

Computers & Industrial Engineering

journal homepage: www.elsevier.com/locate/caie

Scheduling operating rooms with consideration of all resources, post anesthesia beds and emergency surgeries

Guillermo Latorre-Núñez [a], Armin Lüer-Villagra [a,d], Vladimir Marianov [a,*], Carlos Obreque [b], Francisco Ramis [b], Liliana Neriz [c]

[a] Department of Electrical Engineering, Pontificia Universidad Católica de Chile, Santiago, Chile
[b] Department of Industrial Engineering, Universidad del Bío-Bío, Concepción, Chile
[c] Department of Management and Information Systems, Universidad de Chile, Santiago, Chile
[d] Department of Engineering Sciences, Universidad Andres Bello, Santiago, Chile

ARTICLE INFO

Article history:
Received 30 July 2014
Received in revised form 28 September 2015
Accepted 10 May 2016
Available online 11 May 2016

Keywords:
Operating room scheduling
Emergency surgeries
Hybrid flow shop
Constraint programming
Genetic algorithm

ABSTRACT

Surgery rooms are among the most expensive resources in hospitals and clinics. Their scheduling is difficult because, in addition to the surgical room itself, each surgery requires a particular combination of human resources, as well as different pieces of equipment and materials. Furthermore, after each surgery, a post-anesthesia bed is required for the patient to recover. Finally, in addition to planned surgeries, the scheduling must be made in such a way as to accommodate the emergency surgeries that may arrive during each day, which must be attended within a limited time. We address the surgery scheduling problem considering simultaneously, for the first time, the operating rooms, the post anesthesia recovery, the resources required by the surgery and the possible arrival of emergency surgeries. We propose an integer linear programming model that allows finding optimal solutions for small size instances, we transform it to use constraint programming, and develop a metaheuristic based on a genetic algorithm and a constructive heuristic, that solves larger size instances. Finally, we present numerical experiments.

1. Introduction

The relevance of the problem of planning and scheduling of operating rooms (ORs) comes from the fact that ORs are among the most expensive and fundamental resources of a hospital (Cardoen, Demeulemeester, & Beliën, 2010; Guerriero & Guido, 2011). Among other consequences, poor scheduling may generate idle time intervals, an excess of overtime, and delays or cancellations of surgeries, that are reflected in excess cost and loss of revenue for the hospital.

The process of scheduling of surgeries in ORs has two main stages: the daily assignment of patients and the appropriate sequencing of the surgeries (Cardoen, Demeulemeester, & Beliën, 2009a; Guerriero & Guido, 2011). The first stage consists of determining the set of patients that should be operated on a particular day over a given planning horizon (Guinet & Chaabane, 2003). Patients that are candidates for surgery are generally selected from patient waiting lists that are generated by the hospital (Persson & Persson, 2009). This selection is based on resource availability, surgery priority, and the patient's waiting time in the waiting list. At this point, a surgeon is assigned to the surgery. In the second stage, the daily programming of each operating room is prepared, which is known as the Surgical Table Programming or Operating Table (OT). This activity determines the sequencing of the surgeries in the ORs and is scheduled 24–48 h before the event (May, Spangler, Strum, & Vargas, 2011). In this paper, we concentrate on the second step of the process of surgery scheduling, that is the sequencing of surgeries in the ORs.

The OT programmer should build the sequence taking into account the ORs availability as well as the physical and human resources needed to perform each surgery. At the same time, the programmer must consider the availability of a post-anesthesia recovery bed for the patient, thus making sure that the surgical room is not blocked, which would impact the following surgery. Not considering any of these physical or human resources may lead to the cancellation of scheduled surgeries.

Normally, the OT programmer is a physician or a nurse, familiar with the process of each of the scheduled surgeries. Nevertheless, even though the programmer is familiar with the elements, time and care needed to perform the surgery, achieving a good schedule is a very difficult process when a large number of factors need to be simultaneously addressed. In addition, every day, a number of

* Corresponding author.
E-mail address: marianov@ing.puc.cl (V. Marianov).

http://dx.doi.org/10.1016/j.cie.2016.05.016
0360-8352/

emergency surgeries arrive to the hospital which must be handled as soon as possible. Thus, even when surgeons are very skilled in performing surgery, large amounts of idle time occur due mainly to poor scheduling.

Our main contribution is the construction of the OT simultaneously considering for the first time the operating rooms, resources, post-anesthesia recovery beds, and emergency surgeries. We propose different tools to solve the problem and assess the results using computational tests.

The remainder of the article is organized as follows: Section 2 reviews the literature. Section 3 describes the problem. Section 4 presents the exact solution approaches and the tools proposed to solve of the problem. Section 5 describes the heuristic. Section 6 presents and describes the computer experiments. Finally, in Section 7 we offer conclusions and outline directions for future work.

2. Literature review

The scheduling of ORs is a broad topic. A detailed review can be found in Cardoen et al. (2010) and Guerriero and Guido (2011). Here, we focus on the literature that is directly related. Among the relevant papers are Jebali, Hadj Alouane, and Ladet (2006) and Dekhici and Belkadi (2010), in which the assignment and sequencing of surgeries are treated as a hybrid flow shop (HFS), without considering any of the required additional resources in the scheduling process. Roland, di Martinelly, and Riane (2006) and Roland, Di Martinelly, Riane, and Pochet (2010) emphasize the importance of considering additional resources, specially human, in the surgery scheduling process but the post-anesthesia recovery stage is not taken into account.

In Pham and Klinkert (2008), each surgical intervention is described as a predetermined sequence of activities with a maximum allowed waiting time between two consecutive activities, with a set of resources being assigned to each of the activities. The authors develop a mixed linear integer programming (MILP) model. Vijayakumar, Parikh, Scott, Barnes, and Gallimore (2013) consider the resources in the problem and use a dual bin-packing problem model. Given the number of days and resources available for the schedule, they maximize the number of surgeries to schedule. They model the problem as a MILP and develop a heuristic to solve the problem without considering the second stage (recovery). The surgical working affinity group (group of surgeons, nurses and personnel that usually work together) is included as a limitation in Meskens, Duvivier, and Hanset (2013), and constraint programming (CP) is used to solve the problem. Lee and Yih (2014) determine the start times of the surgeries in the operating room based on the availability of beds in the unit Post-Anesthesia Care Unit (PACU) and the uncertainty of the duration of tasks. The problem is approached as a flexible job shop with fuzzy times. To solve it, they develop a genetic algorithm that determines the order of the surgeries and a heuristic to determine the start times.

Emergency surgeries cannot be programmed in advance, since they may arrive at any moment of the day. However, they must be attended promptly. To address this type of uncertainty, Erdem, Qu, and Shi (2012) formulate a MILP model and a genetic algorithm that allows rescheduling elective surgeries. The proposed model minimizes the cost of postponing surgeries, the overtime, and the cost of rejecting an emergency patient by deriving her to a different hospital. Wullink et al. (2007) use simulation to evaluate reservation of time for emergency surgeries. They evaluate two traditional methodologies: (a) concentrating all reserve capacity into an ORs dedicated exclusively to emergency surgery, and (b) uniformly reserving capacity in all ORs dedicated to elective surgery. The performance measures are waiting time, overtime, and OR utilization. They conclude that the second approach is the most efficient. van Essen, Hans, Hurink, and Oversberg (2012) study how to minimize the emergency surgery waiting time, for which they introduce two new concepts: the ‘‘break-in-moment” (BIM) that corresponds to the exact moment in which an emergency surgery can be performed after the conclusion of an elective surgery, and the ‘‘break-in-interval” (BII), defined as the interval between two consecutive BIMs. What they look for is distributing BIMs uniformly throughout the day, for which they minimize the maximum BII. The authors assume that surgeries are programmed without dead times in between. In all these works, they use the fact, recognized in the literature, that there exists a so-called ‘golden hour’: a one-hour period within which care must be provided to trauma emergency patients (Fleet & Poitras, 2011; Newgard et al., 2010).

We synthesize the main literature in the following Table 1, summarizing the related works. The first column lists the authors, while the second to seventh columns show the features of the studied problem. As Table 1 shows, most of the papers partially address the features involved in the problem. Our model covers most of the features: programming for minimum makespan and consideration of the waiting time of emergency surgeries, sequencing and individual assignment of all the medical staff as well as all the resources, and consideration of the two important stages: the OR and the recovery bed.

3. Problem description

A patient's surgery process consists of two main stages. The first stage makes use of the surgery room and it is divided into the OR preparation for surgery, the surgery itself (surgical act), and the cleaning of the OR. The second stage is the post-anesthesia recovery of the patient that takes place in a recovery room.

In general, ORs have different sizes and characteristics, thus a surgery can only be assigned to an operating room the meets the conditions required to perform the operation. Priorities among surgeries may also occur. For example, it may be necessary to move towards the end of the day those surgeries that may contaminate the room to a larger degree.

Each surgery requires a previously assigned physician or surgeon. The surgeon is only present during the surgery act and requires a certain amount of time between interventions (mainly for cleaning, change of clothes, and rest) that depends on the duration of the previously performed surgery.

During a surgery additional resources are required, often scarce. These resources may be nurses, anesthesiologists and other professionals, or physical resources (instruments, imaging equipment, etc.). One or more units of a particular type of resource (human or physical) may be available. Each resource unit assigned to a surgery must be available throughout the preparation and performance of the surgery and has a fixed preparation time that depends on the resource type.

Once the surgery is finished, the patient must be transported to a bed of PACU. Some patients, after surgery, require special care and must be transported to an Intensive Care Unit (ICU). Thus, post-anesthesia recovery may require a special bed.

In addition, demand for emergency surgeries may occur at any time and must be attended within a maximum waiting time.

The objective of the problem is to minimize the closing time of the last operating room in use (makespan).

4. Proposed exact approaches

The problem of scheduling operating rooms, resources and post anesthesia care units can be seen as a hybrid flow shop (HFS) scheduling problem. This is an NP-Hard combinatorial problem