A pre-assignment heuristic algorithm for the Master Surgical Schedule Problem (MSSP)

Article *in* Annals of Operations Research · July 2010

DOI: 10.1007/s10479-009-0568-6 · Source: DBLP

2 authors:

Elena Tanfani
Università degli Studi di Genova
61 PUBLICATIONS **1,046** CITATIONS
SEE PROFILE

Angela Testi
Università degli Studi di Genova
68 PUBLICATIONS **778** CITATIONS
SEE PROFILE

Ann Oper Res (2010) 178: 105–119
DOI 10.1007/s10479-009-0568-6

A pre-assignment heuristic algorithm for the Master Surgical Schedule Problem (MSSP)

Elena Tànfani · Angela Testi

Published online: 20 May 2009

Abstract In this paper a 0–1 linear programming model and a solution heuristic algorithm are developed in order to solve the so-called Master Surgical Schedule Problem (MSSP). Given a hospital department made up of different surgical units (i.e. wards) sharing a given number of Operating Rooms (ORs), the problem herein addressed is determining the assignment among wards and ORs during a given planning horizon, together with the subset of patients to be operated on during each day. Different resource constraints related to operating block time length, maximum OR overtime allowable by collective labour agreement and legislation, patient length of stay (LOS), available OR equipment, number of surgeons, number of stay and ICU beds, are considered. Firstly, a 0–1 linear programming model intended to minimise a cost function based upon a priority score, that takes into proper account both the waiting time and the urgency status of each patient, is developed. Successively, an heuristic algorithm that enables us to embody some pre-assignment rules to solve this NP-hard combinatorial optimisation problem, is presented. In particular, we force the assignment of each patient to a subset of days depending on his/her expected length of stay in order to allow closing some stay areas during the weekend and hence reducing overall hospitalisation cost of the department. The results of an extensive computational experimentation aimed at showing the algorithm efficiency in terms of computational time and solution effectiveness are given and analysed.

Keywords Operating room planning · 0–1 linear programming model · Heuristic algorithm · Computational results

E. Tànfani (✉) · A. Testi
Department of Economics and Quantitative Methods (DIEM), University of Genova, Via Vivaldi 5, 16126 Genova, Italy
e-mail: etanfani@economia.unige.it

A. Testi
e-mail: testi@economia.unige.it

1 Introduction and literature review

Healthcare providers, either in public or in private sectors, are facing increasing pressure to improve cost efficiency and productivity. A critical task in these services is planning the Operating Rooms (ORs) so that hospital resources are efficiently allocated and patient satisfaction is assured. This problem is not new from an economic point of view, i.e. allocating a given amount of resources the best way possible. In general, as in the case study herein analysed, different surgical units (i.e. wards) demand ORs to treat elective patients, each therefore competing for a limited supply of OR blocks of time (i.e. sessions). Note that a ward is considered as being a self-contained service equivalent to a sub-speciality in other countries. The question is, how can demand and supply meet? How should available OR sessions be planned to attain overall efficiency? The OR planning problem herein addressed thus regards the operational decisions that follow tactical ones aimed at determining the amount of OR time available (Dexter et al. 2005). Different objectives must be properly taken into account in order to achieve good, effective OR planning. Some of them are: obtaining a balanced distribution of operating time among wards, maximising the utilisation rate, minimising both OR idle time and overtime, maximising throughput, minimising average patient waiting time and so on.

Dealing with the problem of sharing overall OR time across different wards, Dexter and Macario (2002) argue that OR time should be allocated to maximise the number of surgical cases performed. Blake and Carter (2002) propose a methodology that uses two linear goal programming models in order to determine the trade-off among cost, volume and clinical necessity. Dexter et al. (2005) consider the economic consequences of OR planning incorporating financial criteria and uncertainty in a ward's future workload.

From quite a different stand point, other authors deal with the problem of determining the so-called Master Surgical Schedule (MSS), that is a cyclic timetable that determines the ward associated with each OR session and must then be updated whenever the total amount of OR time changes. This can occur not only as a response to long term change in the gross number of stated OR hours, but also in response to seasonal fluctuations in demand. The literature on MSS is rather poor. Blake et al. (2002) propose an integer programming model that minimises the weighted average undersupply of OR hours (i.e. allocating a number of OR hours to each ward as close as possible to its target number of OR hours). Jebali et al. (2006) propose a 0–1 linear programming model aimed at minimising OR overtime and undertime cost as well as hospitalisation costs related to the number of days patients are kept in the hospital waiting for an operation or procedure.

A third problem is scheduling patients in each time block. Guinet and Chaabane (2003) describe this problem as a general assignment problem aimed at reducing patient stay duration and OR overtime costs. Ozkarahan (2000) introduces a goal programming model for the assignment of surgical operations among multiple ORs with the aim of preventing over or under capacity loading, while Sier et al. (1997) use simulated annealing to find improved solutions for the scheduling of surgical procedures in hospitals. Testi et al. (2007) propose a three-phase approach that has proven to be effective in improving OR productivity by increasing the number of operations performed as well as reducing overruns and shifted operations. Bowers and Mould (2004) use a simulation model in order to examine a policy of including planned elective patients within the orthopaedic trauma theatre sessions, showing that the utilisation of trauma sessions can be increased by scheduling elective patients willing to accept the possibility of their treatment being cancelled.

As far as the authors know, the three problems above described have never been approached as a whole, in a concise solution approach. In this paper, we would like to supply

surgeons with an analytical tool for planning surgical activity in order to determine: (1) how many OR sessions out of the total available in a hospital should be assigned to each ward and on which day of the week (MSS); (2) how many and who the patients are to be selected from the waiting list in order to be operated on in each OR session of the planned MSS determined sub (1).

Moreover, one of the main innovations of the approach herein presented is the objective of maximising overall societal benefit, i.e. improving patient satisfaction by reducing weighted waiting times and also improving hospital efficiency by reducing production costs.

The organisation of the paper is as follows; in Sect. 2 the problem formulation is given, while the 0–1 linear programming model for the problem is presented in Sect. 3. In Sect. 4 the heuristic algorithm approach is analysed in detail. An application of our algorithm is given in Sect. 5 by solving an example step by step, while the results of an extensive computational experimentation aimed at validating the proposed approach are reported in Sect. 6. Lastly, some conclusions and outlines for future work are given in Sect. 7.

2 Problem formulation

The problem herein addressed is that of allocating a set O of v OR sessions available in a given planning horizon to a set W of m wards belonging to a given department/hospital, together with determining among a set I of n patients how many and who the patients are who should be scheduled to be operated on during each operating session. Note that the o-th session is actually addressed by indices k, t representing, respectively, the OR(k) and the day of the planning horizon (t). We will denote by K and T, respectively, the set of ORs and days, and by c and b their corresponding cardinality.

The objective function is intended to minimise the sum of retrospective and cross-section weighted waiting times (Sanmartin 2002). Note that, the retrospective measure reports the waited time of the admitted patients computed retrospectively after their admissions. Instead, the cross-sectional index indicates the time already waited by patients still on the list at the end of the planning horizon. Moreover, the novelty of the objective function here introduced is using a prioritisation algorithm for managing patient admission (see Mullen 2003 for a complete review of prioritisation formulas) that weighs the chronological waiting time with the urgency coefficient of the corresponding Urgency Related Group (URG) of each patient (Testi et al. 2008).

The planning decision has to satisfy many resource constraints related to OR session length, maximum OR overtime allowable by the current, collective labour agreement and hospital budget constraints, expected patient length of stay (LOS), OR available equipment, number of surgeons for each ward available to operate on each day, as well as number of Intensive Care Unit (ICU) and stay beds, etc.

We assume that the number of patients on the waiting list is greater than the maximum number of patients who can be operated on during the planning horizon considered; this means that we are concerned with the problem of selecting some patients to be operated on among all of those who are waiting. Moreover, like most papers in this field, in order for solutions to be obtained and applicable in realistic hospital settings, many restrictive assumptions have been made. In what follows, we assume that: (i) the number of ORs and the number and length of OR sessions available for elective surgery in the department are invariable during the planning horizon; (ii) only one ward should be assigned to a surgery room during a given session, i.e. sessions can not be split among wards; (iii) emergency patients and outpatients are not considered to be scheduled since they use extra operating

rooms reserved for them; (iv) uncertainty considerations in estimates of market demand and length of stay duration have been excluded.

Firstly, a 0–1 linear programming model is developed. The model is intended to minimise a cost function based upon a priority score that takes into proper account both the patient waiting time and urgency status. Successively, a heuristic approach, that enables us to embody some pre-assignment rules in order to be able to solve this NP-hard combinatorial optimisation problem, is presented.

For the model formulation the following notation has been used:

- I set of patients, $i \in \{1, 2, \ldots, n\}$ and $|I| = n$;
- W set of wards, $w \in \{1, 2, \ldots, m\}$ and $|W| = m$;
- K set of operating rooms, $k \in \{1, 2, \ldots, c\}$ and $|K| = c$;
- T set of days belonging to the planning horizon, $t \in \{1, 2, \ldots, b\}$ and $|T| = b$;
- O set of sessions, $o \in \{1, 2, \ldots, v\}$ and $|O| = v$;
- I_w subset of patients who belong to ward w, so that $I_w \subseteq I$, $I_w \cap I_h = \emptyset$, $w, h \in W$, $\bigcup_{w=1}^{m} I_w = I$ and $|I_w| \geq |I_{w+1}|$;
- I_h subset of patients who have an expected LOS of h, so that patient i belong to I_h if and only if $\alpha_i = h$ and that $I_h \subseteq I$, $I_w \cap I_h = \emptyset$ and $\bigcup_{h=1}^{6} I_h = I$;
- T_h subset of days when patient i, $\forall i \in I_h$, cannot be operated on;
- d_i elapsed days since referral of patient i;
- p_i operating time of patient i;
- ρ_i urgency coefficient of patient i;
- α_i expected LOS of patient i;
- s_{kt} session length, i.e. number of hours available for surgery in OR k and day t;
- $\bar{s}$ average session length;
- f_{kt} maximum overtime (in hours) available in OR k and day t;
- l_w lower bound on the minimum # of sessions to be assigned to ward w;
- u_w upper bound on the maximum # of sessions to be assigned to ward w;
- e_w # of operating teams/surgeons belonging to ward w;
- g_t # of beds available on day t;
- q_t # of ICU beds available on day t;
- β_i 1 if patient i needs an ICU bed after operation, 0 otherwise;
- N maximum # of patients that can be scheduled during the planning horizon in the case all of them have a operating time of one hour, so that $N = v \cdot \bar{s}$;

3 The 0–1 linear programming model for MSSP

We assume x_{ikt}, y_{wkt} and z_i as decision variables of the problem, with the following specification:

$$x_{ikt} = \begin{cases} 1 & \text{if patient } i \text{ is assigned to OR } k \text{ on day } t \\ 0 & \text{otherwise} \end{cases}$$

$$y_{wkt} = \begin{cases} 1 & \text{if ward } w \text{ is assigned to OR } k \text{ on day } t \\ 0 & \text{otherwise} \end{cases}$$

$$z_i = \begin{cases} 1 & \text{if patient } i \text{ is not selected to be operated on during the planning horizon} \\ 0 & \text{otherwise} \end{cases}$$

An instance pre-processing is made before running the model so that all variables x_{ikt} and y_{wkt} that imply, respectively, the assignment of a patient to an OR where the required equipment is not available and the assignment of a ward to its unavailable days are not generated. Note that session o is actually identified by indices k, t representing, respectively, the OR and the day of the planning horizon, while i identifies the patient. The definition of variables x_{ikt} , y_{wkt} and z_i enables an easy formulation of the underlying model for MSSP.

Model MSSP:

$$\text{Min } Z = \sum_{i=1}^{n}\sum_{k=1}^{c}\sum_{t=1}^{b} x_{ikt}(t-d_i)\rho_i + \sum_{i=1}^{n}[z_i(b+1-d_i)\rho_i] \tag{1}$$

$$\sum_{i\in I_w} x_{ikt} - N y_{wkt} \leq 0 \quad \forall k = 1,2,\ldots,c;\ \forall t = 1,2,\ldots,b;\ \forall w = 1,2,\ldots,m \tag{2}$$

$$\sum_{k=1}^{c}\sum_{t=1}^{b} x_{ikt} + z_i = 1 \quad \forall i = 1,2,\ldots,n \tag{3}$$

$$\sum_{w=1}^{m} y_{wkt} = 1 \quad \forall k = 1,2,\ldots,c;\ \forall t = 1,2,\ldots,b \tag{4}$$

$$\sum_{k=1}^{c}\sum_{t=1}^{b} y_{wkt} \geq l_w \quad \forall w = 1,2,\ldots,m \tag{5}$$

$$\sum_{k=1}^{c}\sum_{t=1}^{b} y_{wkt} \leq u_w \quad \forall w = 1,2,\ldots,m \tag{6}$$

$$\sum_{k=1}^{c}\sum_{t=1}^{b} y_{wkt} - \sum_{k=1}^{c}\sum_{t=1}^{b} y_{w+1kt} \geq 0 \quad \forall w = 1,2,\ldots,m \tag{7}$$

$$\sum_{i\in I_h}\sum_{k=1}^{c}\sum_{t\in T_h} x_{ikt} = 0 \quad \forall h = 2,3,4,5 \tag{8}$$

$$\sum_{i=1}^{n} x_{ikt}\, p_i \leq s_{kt} + f_{kt} \quad \forall k = 1,2,\ldots,c;\ \forall t = 1,2,\ldots,b \tag{9}$$

$$\sum_{i=1}^{n}\sum_{k=1}^{c} x_{ikt} + \sum_{i=1}^{n}\sum_{k=1}^{c}\sum_{a\in T:a<t} x_{ika} \leq g_t \quad \forall t = 2,3,\ldots,b \tag{10}$$

$$\sum_{i=1}^{n}\sum_{k=1}^{c} \beta_i x_{ikt} + \sum_{i=1}^{n}\sum_{k=1}^{c}\sum_{a\in T:a<t} \beta_i x_{ika} \leq q_t \quad \forall t = 2,3,\ldots,b \tag{11}$$

$$\sum_{k=1}^{c} y_{wkt} \leq e_w \quad \forall t = 1,2,\ldots,b;\ \forall w = 1,2,\ldots,m \tag{12}$$

$$x_{ikt} \in \{0,1\} \tag{13}$$

$$y_{wkt} \in \{0,1\} \tag{14}$$

$$z_i \in \{0,1\} \tag{15}$$

(1) is the objective function minimising the sum of the weighted times of all patients, i.e. the sum of the waiting cost of patients scheduled to be operated on during the planning horizon and of those who will remain on the waiting lists. Constraints (2) allow the intermediate binary variable y_{wtk} to take value 1 if some patients belonging to ward w are scheduled in operating room k and day t and 0 otherwise. Constraints (3) force each patient being selected only once at most and allow the variable z_i to take value 1 if patient i is not scheduled to be operated on during the planning horizon. The assignment constraints (4) ensure that an operating room on a given day can be assigned to one ward only, while constraints (5) and (6) are the lower and upper bound constraints that force the total number of sessions assigned to ward w to be at least more or less than some fixed bounds l_w and u_w. Constraints (7) avoid a ward with a bigger waiting list from obtaining fewer sessions than a ward with a shorter one. (8) are the so-called length of stay constraints that avoid a patient with LOS $= h$ from being operated on the subset of days T_h which would not allow him to be dismissed before the weekend. Constraints (9) impose that in operating room k the overtime does not exceed the maximum number f_{kt} of overtime hours available in day t. Constraints (10) and (11) verify that the number of patients who can be operated on each day is limited by stay and ICU bed availability, respectively. Constraints (12) impose that on day t each ward can be assigned to a number of ORs at least equal to the number of operating teams available. Lastly, in (13), (14) and (15) the binary decision variables of the problem are defined.

Note that the model size results in $(ncb + mcb + n)$ variables and $(cbm + n + 2cb + 3m + h + 2b + bm)$ constraints. The model has been tested on 240 instances and the results are given in Sect. 6.

4 The solution heuristic algorithm

The *MSSP* herein addressed is really complicated because of its combinatorial nature. Moreover, by the analysis of the number and type of constraints involved, it appears that the MSSP is NP-hard due to constraints (9) that make it a bin packing—like problem (Wolsey 1999). In order to solve real large setting instances we propose a heuristic algorithm made up of three main phases. The first phase, i.e. the patient selection procedure, is aimed at creating a subset $\overline{I}$ that denotes the updated set of patients who will be considered to be selected to be operated on during the planning horizon. The second phase, i.e. the pre-assignment procedure, is aimed at considering the set of available sessions split into different partitions according to the day of the week and forcing the assignment of each patient to certain days which are established 'a priori' depending on his/her LOS. The third phase, i.e. the instance pre-processing, is aimed at reducing the feasible region of each patient $i \in \overline{I}$. Lastly, a modified version of *MSSP Model* that embodies the results of the pre-assignment and pre-processing phases is used to solve the problem. In this way, selecting the operation day of patient $i \in \overline{I}$ is limited to a subset of days of the planning horizon thus reducing the solution space of the problem and therefore its complexity as well.

Note that from the hospital point of view, the algorithm also allows beds to be split into two separate stay areas: the first (short stay area), where only patients with an expected length of stay less than 5 days are admitted, can be closed during the weekend with relevant cost savings. The second (long stay area) would be where all other patients are admitted. Note that the algorithm can be applied when at least one work week is considered as planning horizon.

Phase 1. Patient selection procedure

Partitioning I We split set I of all patients on the waiting lists into six subsets I_h, according to the expected LOS α_i of patient i, so that $i \in I_h$ if and only if $\alpha_i = h$, $\forall h = 1, \ldots, 6$. This means that patients are grouped together according to their expected length of stay, so that $\bigcup_{h=1,\ldots,6} I_h = I$ and $I_g \cap I_h = \emptyset$, $\forall h \neq g,\ h, g = 1, \ldots, 6$. Elements belonging to I_1 are hence patients who have an expected LOS of one day, whereas elements belonging to I_2 are patients having an expected LOS of two days, and so on. Note that I_6 conventionally represents the subset of all patients with an expected LOS greater than 5 days (long stay patients). The long stay patients can be grouped and managed together because they all require a bed for the weekend and, consequently, will be associated with the long stay area.

Sizing, reducing and sorting I_h, $h = 1, \ldots, 6$ For each subset of patients I_h, $h = 1, \ldots, 6$, we compute the "upper bound" θ_h as a percentage of the maximum number N of patients that can be scheduled during the planning horizon. This percentage is computed as the weighted sum of two components: (i) π_h, that is the percentage of patients on the waiting lists who have an expected LOS equal to h; (ii) δ_h, that is the percentage of the overall priority status or total weighted waiting time of patients belonging to I_h, such that:

$$\theta_h = \left\| (\varepsilon \cdot \pi_h + \eta \cdot \delta_h) \cdot N \right\| \tag{16}$$

where ε and η are positive numbers ($\varepsilon + \eta = 1$). At the end of this step, following (16), we know the maximum number of patients belonging to each subset I_h that must be considered to be scheduled during the planning horizon.

We then sort the patients in each subset I_h according to their priority status and assign to $\overline{I}_h$ the first θ_h patients, so that $\bigcup_{h=1,\ldots,6} \overline{I}_h = \overline{I}$.

Computing $\chi_h, h = 1, \ldots, 6$ Starting by the value of θ_h we, firstly, compute the corresponding "upper bound" Ψ_h of the number of sessions required to operated on all patients belonging to $\overline{I}_h$, as follow:

$$\psi_h = \frac{\theta_h p_h^*}{\bar{s}} \tag{17}$$

where p_h^* is the average expected operating time of patients belonging to I_h, while $\bar{s}$ is the average session length. Then, we compute the "effective" integer number of sessions χ_h to be assigned to $\overline{I}_h$, $\forall h = 1, \ldots, 6$, among the v available, as follow:

$$\chi_h = \left\| \frac{\psi_h}{\sum_{h=1}^{6} \psi_h} \cdot v \right\| \tag{18}$$

Phase 2. Pre-assignment Procedure

Partitioning O Assuming that we consider a working week of 5 days, we split set O of all available sessions as 5 different subsets O_j, $j = 1, \ldots, 5$, such that $j = 1 = \text{Monday}$, $j = 2 = \text{Tuesday}, \ldots,\ j = 5 = \text{Friday}$. This means that sessions are grouped together according to the day of the week so that $\bigcup_{j=1,\ldots,5} O_j = O$ and $O_g \cap O_h = \emptyset$, $\forall h \neq g,\ h, g = 1, \ldots, 5$. Note that the number of sessions belonging to each subset depends both on the number of ORs as well as the number of days of the planning horizon considered.

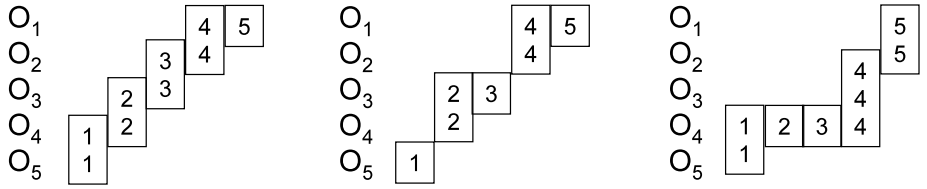

Fig. 1 Examples of different session pre-assignments

Generate $O_{I_h}, h = 1, \ldots, 6$ Successively, we associate subset $\overline{I}_h$, $h = 1, \ldots, 6$, with sessions $o \in O_j$, $j = 1, \ldots, 5$ depending on the value of h and the number χ_h of session required, and generate subset O_{I_h} of all sessions devoted to operate on patient i, $\forall i \in \overline{I}$. The aim is to make the operation of patients out of their pre-defined days unfeasible, thus reducing the decision space for each variable of the problem. In particular, the pre-assignment rules consist of first assigning $\overline{I}_1 \neq \emptyset$, that is to the so-called day surgery patients, to sessions $o \in O_5$ that correspond to the last day of the week, i.e. Friday, and generate $O_{I_1} = O_5$. Then, if $\chi_1 \leq |O_{I_1}|$, the assignment is accepted, since no further sessions are required for operating patients belonging to $\overline{I}_1$. In this case, we start with the search of sessions for operating patients belonging to $\overline{I}_2$; otherwise, we assign $\overline{I}_1$ also to sessions $o \in O_4$, update O_{I_1}, and check the feasibility of the assignment as before. If χ_1 is still greater than $|O_{I_1}|$, we go ahead by adding to O_{I_1} also sessions $o \in O_3$ and so on until $\chi_1 \leq |O_{I_1}|$. Let $\gamma(O_{I_h})$ be the maximum number of patients who can be operated on during sessions $o \in O_{I_h}$, such that $\gamma(O_{I_h}) = \frac{|O_{I_h}| \cdot \overline{s}}{p_h^*}$, and denote by ϕ_h the residual number of patients who can actually be assigned to O_{I_h} so that $\phi_h = \gamma(O_{I_h}) - \frac{\chi_h \cdot \overline{s}}{p_h^*}$. If $\phi_1 > 0$ and $O_{j<5} \in O_{I_1}$, we start by assigning to I_2 the sessions $o \in O_j$ that correspond to the last subset assigned to I_1 initialise O_{I_2} and check the feasibility of the assignment as before; otherwise, if $\phi_1 = 0$, we assign to I_2 the first subset $O_j \notin O_{I_1}$. If $\chi_2 \leq |O_{I_2}|$ the assignment is accepted, otherwise we assign to I_2 also sessions $o \in O_{j-1}$ and so on, updating at each assignment subset O_{I_2}. When O_{I_2} is large enough to operate all patients belonging to I_2 the current assignment is accepted, ϕ_2 is computed and the session assignment procedure will proceed considering set I_3 and so on for all subsets $I_h \neq \emptyset$, $h = 1, \ldots, 5$.

Lastly, if $I_6 \neq \emptyset$ we assign to I_6 set O of all available sessions, so that $O_{I_6} = O$. This means that long stay patients can be operated on all days of the week, since they being required to be kept in the hospital during the weekend (long stay area). At the end of this step, we then have set O_{I_h}, $h = 1, \ldots, 6$, which identifies all sessions where patients belonging to $\overline{I}_h$, $\forall h = 1, \ldots, 6$ can be operated on.

A *c*-like scheme of the algorithm for the pre-assignment phase is reported in the Appendix. Note that this procedure follows the basic criterion by which short stay patients, i.e. patients with an expected LOS less than 5 days, can be operated on only on those days of the week which allow them to be dismissed before the weekend (short stay area) and, consequently, makes redundant constraints (9) which can then be relaxed.

Some examples of different session pre-assignments to short stay patients, i.e. with LOS equal to $1, 2, \ldots, 5$, are reported in Fig. 1.

Phase 3. Instance pre-processing

Following the solution obtained by the pre-assignment procedure, for each patient $i \in \overline{I}_h$ we remove from set O all potential operating sessions that a priori are not to be considered

for operating patient i. In particular, we initially assign set O to $\overline{O}$, where $\overline{O}$ denotes the updated set of available sessions $o_{kt} \in O_{I_h}$, to operate on patient $i \in \overline{I}_h$, and remove from $\overline{O}$ those sessions o_{kt} that correspond to a day of the week that do not correspond to the session pre-assignment defined in Phase 2. Consequently, all variables x_{ikt} so that $o \notin O_{I_h}, \forall i \in \overline{I}_h$, are not generated.

It can be easily noted that the time complexity of the overall algorithm is bounded by a quadratic polynomial function in the number of sessions; in particular, the worst case complexity is $O(hv^2)$.

The modified model Lastly, we solve the *MSSP model* according to the modifications performed in the previous phases, relaxing constraints (9) and considering set $\bar{O}$ instead of the original set O of sessions with subsets $\overline{I}_h$, $h = 1, \ldots, 6$, and their corresponding feasible session assignments O_{I_h}.

5 A sample example

To give an idea of how the proposed approach for MSSP works, let us present a medium-sized case study. We are involved with a hospital department where 80 patients, belonging to 2 wards, with different urgency status (URG), LOS and operating times are waiting for surgery. The planning horizon is 5 days and 2 ORs are available for the whole department ($v = 10$). The session length is fixed at 5 hours (included 1 hour of overtime).

Following phase 1 of our algorithm, we, firstly, split set I into six subsets, one for each LOS. We therefore assume that $\varepsilon = 0,5$ and $\eta = 0,5$ and, according to (16)–(18), we compute the upper bounds θ_h of the number of patients belonging to I_h who can be selected to be operated on and the number of sessions χ_h to be assigned to patients having LOS h, $\forall h = 1, \ldots, 6$ (Table 1).

Secondly, we sort the patients belonging to each subset I_h according to their priority status, i.e. their weighted waiting time, and by using the corresponding ordered list (Table 2), we assign to $\overline{I}_h$ the first θ_h patients, $\forall h = 1, \ldots, 6$.

Successively, by applying the session pre-assignment procedure (Phase 3), we obtain subsets O_{I_h}, $h = 1, 2, \ldots, 6$, given by: $O_{I_1} = \{O_5, O_4\}$, $O_{I_2} = \{O_4, O_3\}$, $O_{I_3} = \{O_3, O_2\}$, $O_{I_4} = \{O_2, O_1\}$, $O_{I_5} = \{O_1\}$, $O_{I_6} = \{O_5, O_4, O_3, O_2, O_1\}$.

Figure 2, reports the MSS obtained by solving the modified *MSSP Model* that gives the assignment between wards and ORs for each day of the week. Note that, during the considered week, 30 patients are admitted to be operated on and a total overtime of 7.5 hours is needed.

Table 1 Maximum number of patients to be selected and sessions needed for each LOS

LOS(h)	π_h	δ_h	θ_h	p_h^*	Ψ_h	$\frac{\Psi_h}{\Psi}$	χ_h
1	0.2	0.27	12	2.4	5.8	0.23	2
2	0.2	0.25	11	2.7	5.9	0.23	2
3	0.2	0.16	9	2.5	4.5	0.18	2
4	0.2	0.16	9	2.7	4.9	0.19	2
5	0.1	0.08	5	2.2	2.2	0.09	1
6	0.1	0.05	4	2.8	2.2	0.09	1
Total	1.0	1.0	50.0		25.5	1.0	10

Table 2 Patient characteristics and ordered list of the sample example

LOS	URG	Ward	Priority	Ordered list	LOS	URG	Ward	Priority	Ordered list
1	A1	*1*	1800	1	3	B	*1*	240	9
1	A1	*1*	1575	2	3	B	*2*	228	10
1	A1	*2*	945	3	3	B	*1*	210	11
1	A1	*2*	675	4	3	A2	*1*	120	12
1	A1	*2*	540	5	3	C	*2*	74	13
1	A1	*2*	504	6	3	C	*1*	44	14
1	A1	*1*	456	7	3	D	*1*	34	15
1	A2	*1*	300	8	3	D	*2*	30	16
1	A2	*1*	264	9	4	A1	*1*	1890	1
1	A2	*1*	240	10	4	A1	*1*	900	2
1	B	*1*	204	11	4	A1	*1*	540	3
1	C	*1*	84	12	4	A2	*1*	456	4
1	C	*1*	48	13	4	A2	*1*	444	5
1	C	*2*	40	14	4	A2	*1*	420	6
1	B	*1*	30	15	4	A2	*2*	360	7
1	C	*1*	20	16	4	B	*1*	222	8
2	A1	*2*	1800	1	4	B	*2*	210	9
2	A1	*1*	1665	2	4	B	*1*	180	10
2	A1	*1*	1350	3	4	B	*2*	180	11
2	A1	*1*	540	4	4	A2	*2*	120	12
2	A1	*1*	504	5	4	D	*2*	25	13
2	A1	*2*	450	6	4	D	*1*	12	14
2	A1	*2*	360	7	4	D	*1*	10	15
2	A1	*2*	264	8	4	D	*2*	10	16
2	A2	*1*	252	9	5	A1	*1*	2025	1
2	A2	*1*	180	10	5	A1	*1*	420	2
2	B	*1*	144	11	5	A2	*1*	360	3
2	C	*1*	90	12	5	A2	*2*	300	4
2	C	*2*	80	13	5	B	*2*	204	5
2	C	*2*	70	14	5	B	*1*	60	6
2	B	*1*	60	15	5	D	*1*	30	7
2	C	*1*	60	16	5	D	*2*	20	8
3	A1	*2*	1575	1	6	A1	*1*	1350	1
3	A1	*1*	1125	2	6	A2	*1*	360	2
3	A2	*2*	540	3	6	A2	*2*	240	3
3	A2	*1*	480	4	6	B	*1*	150	4
3	A1	*1*	450	5	6	A2	*1*	120	5
3	A2	*2*	444	6	6	D	*2*	42	6
3	A2	*1*	360	7	6	D	*1*	34	7
3	A2	*1*	264	8	6	B	*2*	30	8

	Monday	Tuesday	Wednesday	Thursday	Friday
OR1	Ward2	Ward1	Ward1	Ward1	Ward1
OR2	Ward1	Ward1	Ward2	Ward2	Ward2

Fig. 2 MSS of the sample example

The formulation of the problem according to *MSSP Model* results in 900 variables and 162 constraints. The optimal solution, corresponding to the objective function value (1), is $Z^* = 36116$. While using the heuristic approach presented in Sect. 4 the minimum total weighted waiting time is $Z = 36504$, corresponding to an optimality gap of 1.06%. The corresponding computational times are CPU* = 113.83" and CPU = 1.82", respectively, obtained on a PC Pentium IV, using MPL (Maximal Software 2000) and Cplex, together with the heuristic procedures written in Visual C language.

6 Computational experiments

In this section we present some computational experiments aimed at showing the performance of our algorithm. In particular, our test problems are related to a medium-sized department composed of two wards that share two or three ORs open five days a week for five hours a day. Here we test our approach, looking for the MSS of 240 instances that differ from each other pertaining to: (1) the number of patients to be operated on, ranging from 80 to 200; (2) the average urgency status (URG) of patients; (3) the number of ORs; and (4) the number of overtime hours available for each operating session.

To support the examination of algorithm efficiency, all test problems are grouped based on three different overtime availabilities that are either 1, 2 or 3 hours for each session. For each overtime availability 4 test problem sets, characterised by a different number of patients to be operated on (i.e. 80, 100, 160, 200), are generated. Each problem is then solved assuming 2 or 3 ORs. Note that each instance value is the average of 10 problems which differ from each other in the urgency status of patients on the waiting lists. All computational experiments have been performed on the same platform as previously. It is worth mentioning that in the experimentation herein reported the bed capacity constraints (10)–(11) are not integrated in the model solutions since, as observed in the real practice of many surgical departments, beds are not scarce resources and do not constitute the bottleneck of OR planning process.

Tables 3, 4 and 5 reports the results of the instances analysed. The headings are as follows: columns *#ORs* and *#patients* specify the number of ORs available and the number of patients to be operated on for each class of instances; Z^* is the optimal objective function value (1) (i.e. the solution of *Model MSSP*) and Z is the same value obtained by applying our proposed approach to the problem resolution. *CPU** and *CPU* are the computational times, in seconds, corresponding to the above solutions.

Finally, columns Δ_{opt} report the optimality gaps as percentage differences between the optimal solution (Z^*) and the heuristic one (Z) and the corresponding CPU times, given by the percentage ratios $\frac{Z-Z^*}{Z^*}$ and $\frac{|CPU-CPU^*|}{CPU^*}$, respectively.

It is interesting to analyse the difference between the values of the optimal objective function and those corresponding to the MSS obtained with our heuristic algorithm, that is on average equal to 371.7 corresponding to an average optimality gap of 0.73%. Note that the gap between the optimal solution and the heuristics one decreases when the number of patients to be operated on increases. This means that the efficiency of the heuristics algorithm

Table 3 Comparison of solution quality and CPU time—1 hour of session overtime availability

Instance		Objective function (Z)			Computational time (sec)		
# ORs	*# patients*	Z^*	Z	Δ_{opt} (%)	CPU*	CPU	Δ_{opt} (%)
2	80	31516	31786	0.86	156.23	16.22	89.62
	100	46187	46543	0.77	234.45	25.13	89.28
	160	63186	63345	0.25	567.60	43.12	92.40
	200	92857	93045	0.20	765.26	45.24	94.09
	Avg	58436.50	58679.75	0.52	430.89	32.43	91.35
3	80	30576	30823	0.81	189.34	18.17	90.40
	100	45045	45412	0.81	456.12	20.17	95.58
	160	62353	62678	0.52	785.40	41.46	94.72
	200	91547	91896	0.38	823.54	43.12	94.76
	Avg	57380.25	57702.25	0.63	563.60	30.73	93.87

Table 4 Comparison of solution quality and CPU time—2 hours of session overtime availability

Instance		Objective function (Z)			Computational time (sec)		
# ORs	*# patients*	Z^*	Z	Δ_{opt} (%)	CPU*	CPU	Δ_{opt} (%)
2	80	30854	31219	1.18	156.23	21.40	86.30
	100	45012	45367	0.79	234.45	34.31	85.37
	160	62978	63234	0.41	601.23	46.34	92.29
	200	92458	92658	0.22	823.45	47.00	94.29
	Avg	57825.50	58119.50	0.65	453.84	37.26	89.56
3	80	30195	30421	0.75	221.40	22.40	89.88
	100	44367	44680	0.71	267.50	28.50	89.35
	160	62054	62432	0.61	657.30	37.40	94.31
	200	91156	91367	0.23	1178.50	52.30	95.56
	Avg	56943.00	57225.00	0.57	581.18	35.15	92.28

increases when the size of the problem grows, thus indicating that effective solutions can be found even for large instances which cannot be solved by the *MSSP Model* in reasonable computational times.

As expected, the computational time in the case of *MSSP Model* grows noticeable with the number of patients to be operated on, because of the NP-hard nature of the MSSP. Instead using our heuristic procedure, they are always less than 1 minute, corresponding to an average optimality gap of 91.64%.

Note that, problems with 3 OR and up to 200 patients (worst case in Table 5) can be solved to optimality in 1578 seconds, that is 25 minutes approximately. Even if, by the department point of view this time can be considered as reasonable, a method able to find a good suboptimal solution in about 50 seconds is utmost preferable. In particular, in the real practice of a generic hospital department some "last minute" situations could occur at any time thus requiring as fast as possible rescheduling and reassignment of patients. Moreover, the reason of implanting an algorithm for reducing the CPU times is prevalently due to the possibility of making tractable larger instances with more wards, ORs and patients in the waiting lists.

Table 5 Comparison of solution quality and CPU time—3 hours of session overtime availability

Instance		Objective function (Z)			Computational time (sec)		
# ORs	*# patients*	Z^*	Z	Δ_{opt} (%)	CPU*	CPU	Δ_{opt} (%)
2	80	30456	30670	0.70	123.50	12.40	89.96
	100	44765	45023	0.58	198.40	25.13	87.33
	160	62674	62965	0.46	598.30	43.12	92.79
	200	92009	92356	0.38	1345.60	45.24	96.64
	Avg	57476.00	57753.50	0.53	566.45	31.47	91.68
3	80	29976	30124	0.49	189.34	18.17	90.40
	100	43985	44219	0.53	456.12	20.17	95.58
	160	61723	62132	0.66	785.40	41.46	94.72
	200	90526	90896	0.41	1578.40	43.12	97.27
	Avg	56552.50	56842.75	0.52	752.32	30.73	94.49

It is worth mentioning that the sister ward of the department which provided us with the data, takes approximately one to one and a half hours to manually fill out similar sized master surgical schedules.

7 Concluding remarks

In this paper a 0–1 linear programming model and a solution heuristic algorithm are developed for solving the Master Surgical Schedule Problem (MSSP). Specifically, the heuristic algorithm implements some pre-assignment rules for assigning patients only to subsets of days depending on their expected length of stay. The solution algorithm has been tested by solving 240 generated instances using realistic parameter values. Computational results indicate that the algorithm can solve all test problems in the magnitude of 50 seconds and with an optimality gap less than 1%. In particular, the most important consideration pertaining to the performance of our heuristic algorithm is the possibility of finding MSSs for large departments with more than 200 patients on the waiting list. Therefore, we believe that the proposed approach is of utmost value and that one of the future directions of this research should be its application on the evaluation of the impact of different strategies aimed at reducing hospital costs and improving hospital productivity.

Acknowledgements The authors wish to thank the anonymous referees for their valuable comments that help in improving the quality of the paper.

Appendix: Procedure pre-assignment

begin

/* initialisation */

h belongs to set $\{1, 2, \ldots, 6\}$ /* LOS length of stay */

j belongs to set $\{1, 2, \ldots, 5\}$ /* Monday $= 1$, Tuesday $= 2$, $\ldots$, Friday $= 5$ */

For each j define $O_j = \{o \in O / j \in (1, 2, \ldots, 5)\}$

For each h set $O_{I_h} = \emptyset$

$A = 0$; /* set to zero counter variable A*/

$B = 1$ /* initialisation of counter variable B */
$\mathrm{LD} = 0$; /* set to zero counter variable A*/

/* Main procedure for short stay patients*/
for $A = 1, 5, ++1$
  if $I_A \neq \emptyset$ **then**
    if $A = 1$ **then**
      do while $(\chi_A > |O_{I_A}|)$
        O_{6-B} is assigned to I_A
        Update $O_{I_A} = O_{I_A} \cup O_{6-B}$
        $\mathrm{LD} = 6 - B$
        $++B$
      endwhile
    else
      if $\phi_{A-1} > 0$ or $I_{A-1} = \emptyset$ **then**
        do while $(\chi_A > |O_{I_A}|)$
          O_{6-B} is assigned to I_A
          Update $O_{I_A} = O_{I_A} \cup O_{6-B}$
          $\mathrm{LD} = 6 - B$
          $++B$
        endwhile
      elseif $\phi_{A-1} = 0$ **then**
        do while $(\chi_A > |O_{I_A}|)$
          $--\mathrm{LD}$
          O_{LD} is assigned to I_A
          Update $O_{I_A} = O_{I_A} \cup O_{\mathrm{LD}}$
        endwhile
      endif
    endif
    /*compute the maximum number of patients who can be operated on during sessions belonging to O_{I_A}*/
    $\gamma(O_{I_A}) = \frac{|O_{I_A}| \cdot \bar{s}}{p_A^*}$
    /* compute the residual number of patients who can be assigned to O_{I_A} */
    $\phi_A = \gamma(O_{I_A}) - \frac{\chi_A \cdot \bar{s}}{p_A^*}$
  else
    $B{+}{+}$
  endif
next A

/* Assign all long stay patients to all sessions */
if $I_6 \neq \emptyset$ **then**
  $O_{I_6} = O$
endif

/* Return O_{I_R}, R belongs to set $\{1, 2, 3, 4, 5, 6\}$ */
for $R = 1, 6, ++1$
  return O_{I_R}
next R
end

References

Blake, J. T., Dexter, F., & Donald, J. (2002). Operating room manager's use of integer programming for assigning block time to surgical groups: A case study. *Anesthesia and Analgesia*, *94*, 143–148.

Blake, J. T., & Carter, M. W. (2002). A goal programming approach to strategic resource allocation in acute care hospitals. *European Journal of Operations Research*, *140*, 541–561.

Bowers, J., & Mould, G. (2004). Managing uncertainty in orthopaedic trauma theatres. *European Journal of Operational Research*, *154*, 599–608.

Dexter, F., & Macario, A. (2002). Changing allocations of operating room time from a system based on historical utilization to one where the aim is to schedule as many surgical cases as possible. *Anesthesia and Analgesia*, *94*, 1272–1279.

Dexter, F., Ledolter, J., & Wachtel, R. E. (2005). Tactical decision making for selective expansion of operating room resources incorporating financial criteria and uncertainty in sub-specialties' future workload. *Anesthesia and Analgesia*, *100*, 1425–1432.

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

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

Mullen, P. M. (2003). Prioritising waiting lists: How and why? *European Journal of Operational Research*, *150*(1), 32–45.

Ozkarahan, I. (2000). Allocation of surgeries to operating rooms using goal programming. *Journal of Medical Systems*, *24*(6), 339–378.

Sanmartin, C. (2002). *Towards standard definitions of waiting times for health care service*. Working paper, Appendix G of the final report of the WCWL (Western Canada Waiting Lists) (pp. 337–372).

Sier, D., Tobin, P., & McGurk, C. (1997). Scheduling surgical procedures. *Journal of the Operational Research Society*, *48*, 884–891.

Testi, A., Tanfani, E., Valente, R., Ansaldo, G. L., & Torre, G. C. (2008). Prioritising surgical waiting list. *Journal of Evaluation in Clinical Practice*, *14*(1), 59–64.

Testi, A., Tanfani, E., & Torre, G. C. (2007). A three phase approach for operating theatre schedules. *Health Care Management Science*, *10*, 163–172.

Wolsey, L. A. (1999). *Integer programming*. New York: Wiley.