Computers & Operations Research 38 (2011) 1452–1463

Contents lists available at ScienceDirect

Computers & Operations Research

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

Local search and lower bounds for the patient admission scheduling problem

Sara Ceschia *, Andrea Schaerf

DIEGM, University of Udine, via delle Scienze 208, I-33100 Udine, Italy

ARTICLE INFO

Available online 12 January 2011

Keywords:
Local search
Simulated annealing
Lower bounds
Healthcare

ABSTRACT

We propose a multi-neighborhood local search procedure to solve a healthcare problem, known as the *Patient Admission Scheduling* problem. We design and experiment with different combinations of neighborhoods, showing that they have diverse effectiveness for different sets of weights of the cost components that constitute the objective function. We also compute many lower bounds based on the relaxation of some constraints. The outcome is that our results compare favorably with the previous work on the problem, improving all available instances, and in some cases are also quite close to the lower bounds. Finally, we propose the application of the technique to the dynamic case, in which admission and discharge dates are not predictable in advance.

1. Introduction

Healthcare is surely one of the most important application domains of optimization in general and of metaheuristics in particular. Many papers have been devoted to healthcare, for example to nurse and physician rostering problems [1,2], and more generally to timetabling problems in hospitals (see, e.g., [3]).

The Patient Admission Scheduling (PAS) problem consists in assigning patients to beds in such a way to maximize both medical treatment effectiveness and patients' comfort. PAS has been defined by Demeester et al. [4], and studied by Bilgin et al. [5] and Demeester et al. [6]. In addition, Demeester maintains a website [7] that publishes the available instances, the current best solutions, and also a solution validator to let other researchers double-check their own solutions. The presence of the validator (a Java `.jar` file, in this case) is very important, as it provides against the risk of misunderstanding the problem formulation and the cost function.

We propose a local search approach to the PAS problem that makes use of different search spaces and neighborhood relations. We also study how to adapt the neighborhood relations for different weights of the components of the cost function. In addition, we propose a relaxation procedure to compute lower bounds (using CPLEX v. 12), which are useful to assess the quality of the solutions more objectively. The outcome of our work is that our results are better than the ones obtained by Bilgin et al. [5], and in some cases also quite close to the lower bounds.

In the PAS problem, it is assumed that admission and discharge dates are known in advance. However, in the actual situations the hospitals have to face these dates may change depending on the evolution of the disease of the patient. Therefore, we also propose a *dynamic* problem, in which admission and discharge dates are not known in advance. We have developed a solver for this case, which is a modification of the one for the *static* case. In practical situations, the static solver is used mainly for simulations and previsional solutions, whereas the dynamic solver is used for the actual day-by-day scheduling.

The paper is organized as follows. In Section 2, we provide the definition of the problem and its mathematical formulation. In Section 3, we discuss related work. In Section 4, we describe our solution technique and, in Section 5, we present the benchmark instances, the lower bounds, and the experimental results. In Section 6 we discuss the dynamic case. Finally, conclusions are drawn in Section 7.

2. Problem definition

The general problem formulation is provided by Demeester et al. [4,6]. We report it here in order to make the paper self-contained. We also describe our preprocessing steps that allow us to improve the efficiency significantly. Finally, we provide the mathematical formulation that we have used to obtain the lower bounds.

2.1. Basic formulation

These are the basic features of the problem:

Day: It is the unit of time and it is used to express the length of the planned stay of each patient in the hospital; the set of

* Corresponding author.
E-mail addresses: sara.ceschia@uniud.it (S. Ceschia), schaerf@uniud.it (A. Schaerf).

0305-0548/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cor.2011.01.007

(consecutive) days included in the problem is called the *planning horizon*.

Patient: She/he is a person who needs some medical treatments, consequently she/he must spend a period in the hospital and she/he should be placed in a bed in a room. Each patient has a *fixed* admission date and discharge date within the planning horizon.

Bed/room/department: A room can be single or can have more beds. The number of beds in a room is called its *capacity* (typically one, two, or four). Patients may (with an extra charge) express preferences for the capacity of the room they will occupy. Each room belongs to a unique department.

Specialism: Each patient needs one or more specific specialisms for her/his treatment. Each department is qualified for the treatment of diseases of various specialisms, but at different levels of expertise. In addition, each specific room has a set of specialisms it is particularly suitable for, each with its level of quality. Levels are expressed in integer values from 1 (highest) to 3 (lowest). Most of the patients need one single specialism for their entire treatment. The patients that need more than one specialism are called *multi-spec* patients; for them, the length of the treatment for each specialism is specified. For example, one patient might need cardiology for the first three days and gerontology for the remaining two.

Room feature: Each room has different features (oxygen, telemetry, ...) necessary to treat particular pathologies. Every bed in a room has the same equipment. Patients may *need* or simply *desire* specific room features.

Room gender policy: Each room has a *gender policy*. The are four different policies, identified by the letters in the set {D, F, M, N}. In rooms with policy F (resp. M) only female (resp. male) patients can be accepted. If the policy is D the room can be occupied by patients of both gender, but on any day the patients in the room must be all of the same gender. Finally, rooms of policy N can be occupied simultaneously by patients of both genders.

Age policy: Some departments are specialized in patients of a specific age range (e.g., pediatrics or gerontology). For these departments there is a limit on the minimum or the maximum age of the patients admitted.

The PAS problem consists in assigning a bed to each patient on each day of her/his stay period. The assignment is subject to the following constraints and objectives:

Bed occupancy (BO): There can be at most one person per bed per day.

Room gender (RG): For each day, the gender of the patients in the same room should obey the gender policy of the room.

Department specialism (DS): The department should have level 1 for the specialism of the patients hosted in the rooms of the department; lower level is penalized as explained in [7].

Room specialism (RS): Similarly to departments, all levels lower than 1 are penalized (see [7]).

Room features (RF): The room should have the features needed or desired by the patients, missing ones are penalized; the penalty is higher for needed ones than for desired ones.

Room preference (RP): Patients should be assigned to rooms of the preferred capacity or smaller.

Patient age (PA): Patients should be assigned to department that can accept patients of their age.

Transfer (Tr): Patients should not change the bed during their stay; a bed change is called a *transfer*. All transfers are penalized in equal way.

The constraint BO is obviously a *hard* constraint, given that the simultaneous assignment of two patients to the same bed clearly makes the solution infeasible. In the current formulation, all the other constraints are considered *soft* and are weighted appropriately. The weights of the various components can be assigned by the final user, based on the specific situation, regulation, and internal policy.

Notice that when a patient needs more specialisms, it is quite acceptable that she/he has to be transferred from one room to another one on the day of the change of treatment. Nevertheless, given that departments and rooms may have more than one specialism, it is also possible that a patient can have all of them in the same bed. For this reason, transfers of multi-spec patients are penalized like all the other transfers. However, our solvers are able to deal also with the general situation, in which the penalty of the transfer is not always the same but depends on the treatment of the patient.

2.2. Preprocessing

Based on the definitions given above, we recognize that the problem can be greatly simplified by means of two preprocessing steps which lead to a new problem formulation.

2.2.1. Room assignment

First of all, it is evident from the formulation that beds belonging to the same room are indistinguishable from each other in terms of features and constraints.

Therefore, we can reformulate the problem as an assignment of patients to *rooms* rather than to *beds*. To this aim, we have to replace BO with the constraint that the number of patients assigned to the same room on each day cannot exceed the capacity of the room.

Given that the output should be delivered in terms of assignments to beds, the room assignment must then be post-processed to be transformed into a bed assignment. In the solution obtained by the post-processor, it is important to avoid to move a patient from one bed to another one in the same room. To this regard, it is easy to prove that if patients are processed on the basis of the order on their admission day, we can always produce an assignment that never moves a patient from one bed to another one in the room.

2.2.2. Patient–room penalty matrix

The second preprocessing step is related to the notions of departments, specialisms, room features, age policy, and preferences. It is evident that all the constraints related to these notions contribute, with their weights, to the penalty of assigning a given patient to a given room.

Therefore, we can "merge" together this information into a single matrix that represents the cost of assigning a patient to a room. This *patient–room penalty matrix* C is computed once for all, when reading the input data, and all the five notions mentioned above (departments, specialisms, room features, age, and preferences) can be removed from the formulation.

The penalty associated to the room gender policy RG can also be *partly* included in the matrix C. More specifically, if the room policy is N there is no penalty, if it is of type F or M, then the penalty of accepting a male patient or a female patient is merged to the matrix C. The only case that is not merged into C, because it depends also on the assignment of the other patients, is the case on policy D, which is the most common one.

2.2.3. Problem reformulation

According to the first preprocessing step, the constraint BO is removed and replaced by the following one:

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

Based on the second preprocessing step, the constraints DS, RS, RF, PA, RP, and RG (for all rooms, but those of policy D) are removed and replaced by the following one.