Operations Research for Health Care 3 (2014) 191–199

Contents lists available at ScienceDirect

Operations Research for Health Care

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

A flexible discrete optimization approach to the physician scheduling problem

Renato Bruni [a,*], Paolo Detti [b]

[a] "Sapienza" Università di Roma, Dip. di Ingegneria Informatica, Automatica e Gestionale, Via Ariosto, 25 - 00185 Roma, Italy
[b] Università di Siena, Dip. di Ingegneria dell'Informazione e Scienze Matematiche, Via Roma, 56 - 53100 Siena, Italy

ARTICLE INFO

Article history:
Received 5 September 2013
Accepted 21 August 2014
Available online 3 September 2014

Keywords:
Staff scheduling
Personnel scheduling
Physicians
Rostering
Mixed Integer Linear model

ABSTRACT

Personnel scheduling deals with the attribution of a number of duty shifts to a number of workers respecting several types of requirements. In this work, the problem of scheduling physicians in health care departments is studied. This problem is NP-hard, and we propose a flexible Mixed Integer Linear Programming formulation that allows easy modifications for representing different situations and scenarios. This formulation can be solved to optimality by a standard Branch-and-Cut procedure even for very long planning horizons. A real-world case study is considered. A comparison of the solutions obtained by the proposed approach with the solutions currently adopted in the considered structure is presented. Results are very encouraging both from the schedule quality (e.g., workload balancing) and from the computational point of view.

1. Introduction

Personnel scheduling problems deal with the assignment of a number of tasks to a number of workers while respecting a number of restrictions that often make the problem harder than a simple assignment problem. Generally this attribution must be done in order to strictly satisfy service requirements and contractual agreements, and to maximize workers' preferences and/or minimize costs, where these objectives are applicable. These problems are very relevant in many working environments and assume different connotations in the different contexts, but typically constitute highly constrained optimization problems. Consequently, in the last few decades they have been studied widely and several different approaches have been proposed (see for references [1,2]). In the field of health care, personnel scheduling problems are particularly pervasive, because many services need to be assured on a continuous basis, twenty-four hours a day, seven days a week, and so any organization of the work must be based on duty rostering.

This work deals with the problem of scheduling physicians in the departments of a health care structure (e.g., hospital), where a schedule is an assignment of physicians to perform different *medical guard services* in those departments. Scheduling health care personnel is in general particularly challenging. However, the quality of these schedules is very important, because high performances in health care services could hardly be achieved when using low quality rosters, e.g., rosters causing unbalanced workloads or excessive stress in the workers. Performing periods of medical guard service in the hospitals is part of the physicians charges. Those periods usually last 10/12 h, hence their good scheduling, taking into account suitable rest periods and any other additional requirement, is essential in improving the overall efficiency of the system.

The problem of personnel scheduling in health care has been addressed by several authors. However, most of the work has been focused on nurse scheduling (see, e.g., [3–6]). The physician scheduling problem is different from nurse scheduling, since, in general, nurses work under collective agreement while physicians may have individual and *ad hoc* contractual duties. Another difference is in the objectives. In the nurse staff problem, both preference satisfaction and costs aspects must be considered. In the physician scheduling, only the satisfaction of the physician is typically taken into account, as physician retention is another issue often faced by hospital administrations. As a consequence, nurse scheduling is usually performed in two steps [7]. First, a schedule is generated to satisfy the collective agreement for the regular staff, minimizing staff shortages and surplus. Then, shortages are solved by using external personnel and overtime. The physician schedule

* Corresponding author.
E-mail address: bruni@dis.uniroma1.it (R. Bruni).

http://dx.doi.org/10.1016/j.orhc.2014.08.003
2211-6923/

is usually a one step process, where scheduling requirements must be joined with physician availability.

Most of works from the literature on physician scheduling deal with the problem of scheduling physicians in particular contexts, such as emergency rooms [8,7,9,10] or operating rooms [11,12]. Recently, in [13], the problem of generating a master schedule for the physicians in a hospital is addressed, by considering the full range of day-to-day activities of the physicians (including surgery, clinics, scopes, calls, administration). However, for large instances, this overall problem could not be solved exactly but only heuristically. In [14], a set covering model for scheduling physicians in a hospital is presented and tested on real-world instances, but only with a 2-week planning horizon. The same instances are solved heuristically in [15]. Despite the differences from nurse and physician scheduling, the mathematical formulations and the solution approaches used for solving the two problems may have points in common. Indeed, the solution approach employed in [16] for the nurse scheduling problem was slightly modified and successfully applied to the physician scheduling problem in [8].

In this work, a Mixed Integer Linear Programming formulation for the physician scheduling problem is presented. This model imposes satisfying all service requirements and contractual agreements, while trying to respect, as far as possible, workers' preferences. The proposed model is able to represent the various aspects of the problem generally considered in literature, and is also flexible, allowing easy modifications for representing different situations. This model can be solved to optimality by means of a standard Branch&Cut procedure for planning horizons even beyond the standard needs. The paper is organized as follows. In the rest of this Section, a brief discussion of the problem complexity is given. In Section 2, the general structure of the problem is formally defined. Section 3 describes the formulation proposed for the general problem, while Section 4 analyzes the capability of this model to represent the different aspects of the problem. Section 5, finally, reports computational results and comparison on real world instances.

1.1. Complexity issues

Since the personnel scheduling problem generally is an assignment of shifts to physicians, in its very basic version it could be modeled as a standard assignment problem, and consequently solved in polynomial time [17]. From the literature, easy cases of personnel scheduling have been modeled as *network flow* models and solved in polynomial time (e.g. [18,19]). However, as soon as the problem gains more richness, more complex models are needed, as, for example, the *multicommodity* network flow models used in the case of railways [20], that are NP-hard [21].

In the case of physician scheduling, a number of additional requirements exist, that make the problem particularly difficult. One of the main complicating requirement is the fact that physicians need to have rest periods after any shift performed. Another complicating requirement is assigning shifts to doctors in order to balance the workload as much as possible. Those requirements imply using additional mathematical machinery for somehow "counting" the number of shifts. Indeed, Brucker et al. [22] show that the physician scheduling problem is NP-hard when the planning horizon is composed by $3n$ shifts, where n is the total number of physicians, and 3 not consecutive shifts must be assigned to each physician (by reduction from the *exact covering by 3-sets* (X3C) problem [17]). Recall that the problem 3-*Satisfiability* (3SAT) reduces to X3C [17], and that the problem qSAT remains NP-complete for any $q \geq 3$. From this, it follows that *exact covering by p-sets* for $p \geq 3$ is NP-complete, too. Therefore, the physician scheduling problem is NP-hard also when the planning horizon is composed by pn shifts and p not consecutive shifts must be assigned to each physician, with $p \geq 3$.

Evidently, the physician scheduling problem has a *combinatorial optimization* structure, with the *ground set* being the set of every possible attribution of shifts to doctors, and each feasible solution is only a *subset* of this ground set producing a certain value for the selected objective (which can be, for example, a measure of the workload balancedness).

2. Definition and general problem structure

In this section, a formal definition of the Physician Scheduling problem is given. We consider the scheduling of physicians to perform different medical guard services in the departments of health care structures such as hospitals. Physicians, hereinafter called also *doctors*, have to guarantee their presence in a number of *departments*, so as to be ready to assist patients and solve the different issues that may arise. This means doctors must cover a number of time slots, or *shifts*, in those departments, according to a number of *rules*. Such rules may be different from case to case, but several general aspects are common to all cases. Basically, there are rules for ensuring *coverage* of the departments, sufficient *rest*, vacation time or other unavailabilities of the doctors, and workload *balancing* among the doctors. Balancing generally constitutes the most important preference aspect in practice, and indeed other preference aspects were not allowed in the analyzed real world case for avoiding disputes. Additional preferences could of course be allowed in specific cases, constituting additional elements in this structure. Therefore, a mathematical model of the problem should provide enough flexibility for introducing them. We now formalize the problem structure, giving also some concrete examples just for helping comprehension. Clearly, there are cases that may considerably differ from such examples, but still fall within the same general structure.

Shifts. The scheduling is generally planned by using a certain *planning horizon*, for instance 6 months or 1 year, and all the shifts in the planning horizon constitute a set $S = \{s_1, \ldots, s_m\}$. This horizon cannot be too short, since doctors need to know their duties with some advance, but not exceedingly long, because otherwise re-organizations, changes in the personnel, etc. would often nullify the rostering. Generally, we need to distinguish among:

1. Shifts located on regular working days, constituting a set $S_d \subset S$. They are for instance shifts lasting the 12 h from 8 a.m. to 8 p.m. of the week days from Monday to Saturday.
2. Shifts located on regular nights, constituting a set $S_n \subset S$. They are for instance shifts lasting the 12 h from 8 p.m. to 8 a.m. following the week days from Monday to Saturday (e.g. Monday night, in common terminology, is the night between Monday and Tuesday, even if Monday officially starts at 0.00 am of the night between Sunday and Monday).
3. Shifts located on Sundays, constituting a set $S_s \subset S$. They are for instance the 12 h day shift and the 12 h night shift located on Sundays.
4. Shifts located on holidays, constituting a set $S_{ho} \subset S$. They are the 12 h day or night shifts that happen to be on a major holiday (Christmas, New Year's day, etc.).
5. Shifts located in weekends, constituting a set $S_{we} \subset S$. They are the 12 h day or night shifts that happen to be on Saturdays or on Sundays.

Clearly, $S_d \cap S_n = \emptyset$, $S_d \cap S_s = \emptyset$, $S_n \cap S_s = \emptyset$, and $S_d \cup S_n \cup S_s = S$. On the contrary, S_{we} and S_{ho} may intersect the above S_d, S_n, S_s, and are needed for evaluating the weight of the