10th International Conference of the Practice and Theory of Automated Timetabling
PATAT 2014, 26-29 August 2014, York, United Kingdom

Personalized nurse rostering through linear programming *

Han Hoogeveen Tim van Weelden
Department of Information and Computing Sciences
Utrecht University
P.O. Box 80089, 3508 TB Utrecht, The Netherlands.
j.a.hoogeveen@uu.nl, timvanweelden@gmail.com

1 Introduction

Our purpose is finding good, personalized rosters for the personnel of the Cardiothoracic Department of UMC Utrecht. The department contains several treatment rooms at different care levels. Taking care of the patients is done 24/7 by approximately 52 nursing employees, each of which is available for a given number of hours, which differs per nurse. There are four types of nurses: student nurses, basic nurses, Medium Care nurses, and Senior nurses. The student nurses are trainees, who work mostly in the day-shift. The basic nurses have a basic qualification only; they can work in all shifts. Next to the basic qualification, the MC and senior nurses have an MC/senior qualification.

The day is divided in three shifts: day (early) shift, late shift, and night shift. For all shifts in the planning period, we know the necessary number of nurses per qualification.

The nurses each have there own preferences regarding their schedule; the hospital tries to obey these as much as possible. Rosters are created per 6-week period. Currently, this is done by hand, since the planning system in use is not able to find reasonable rosters. We present an ILP-based algorithm to solve this problem; it is based on the approach in [2] that was used to solve a simpler problem without qualifications. Moreover, we present a repair heuristic to adapt to changing nurse availability.

Our contribution. We describe an algorithm that can find near-optimal solutions for real-life rostering problems while taking personal preferences into account. This hot topic (in the Netherlands) has not been well-studied in the literature (see [1,2]).

*The working paper this abstract is based can be found at http://www.staff.science.uu.nl/ hooge109/Weelden.pdf

2 Constraints

The goal is to find a feasible roster for each employee such that there are enough nurses with the right qualifications in each shift, and nurse preferences are satisfied as much as possible. Having surplus employees is allowed, preferably well-spread over the day-shifts, but never for night-shifts. Moreover, the presence of student nurses should be evenly divided over the day-shifts.

The individual rosters have to obey several hard constraints. First of all, the number of workshifts should equal the appointment size, but a deviation of 1 is allowed; moreover, every employee should work at most 6 days per week. Next, the night-shifts must be rostered in prespecified blocks. There should be at most 4 night-shifts per 6 weeks, and after a night-shift the nurse must have at least two days off. Furthermore, a nurse must have the same shift on Saturday and Sunday. Moreover, days off must come in blocks of size two or more. Because of healthy rostering, a late-shift cannot be followed by a day-shift. Finally, senior nurses are entitled to two 'quality days'.

Furthermore, there are many soft constraints concerning individual rosters, which are used to determine the quality of the individual rosters. In general, the number of shifts per week should be the same each week, and the night-shifts and weekend-shifts should be well spread over the 6-week period. Examples of personal preferences are to have a regular or occasional day off, to have a specific shift on a day (some nurses perform self-scheduling), to have as many/few late/night shifts as possible, to have as many days off after night-shift as possible, and to have bounds on the number of consecutive shifts.

3 Solution approach

The basic idea (see [2]) is to formulate the problem as an ILP. For each nurse, we construct a set of rosters; each one satisfies all hard constraints issued by both the hospital and the nurse. In constrast to [1], we generate these up to 200.000 rosters per nurse beforehand. For each roster we compute its cost on basis of how well it respects the personal preferences of that nurse. The cost is then scaled to $[0, 1]$ and the ones with cost > 0.5 are removed.

For each roster $s \in S$ we introduce a binary variable x_s indicating whether s is chosen. Roster s contains information as to which nurse it belongs and which shifts it covers. Using these variables, we formulate the constraint that each nurse gets one roster, and that for each combination of a shift and a qualification, we meet the minimum occupancy. We further have variables measuring shortage, surplus, and excessive surplus per shift per qualification; these are penalized in the objective function. The objective is to minimize the total cost of the chosen rosters plus the total shortage, surplus, and excessive surplus penalty.

As the ILP cannot handle these large numbers of variables, we restrict ourselves to a selection of these rosters, for which we then solve the ILP. To make this selection, we solve the LP-relaxation through revised simplex; we put each variable that gets selected in the revised simplex in a column pool. Furthermore, we add variables to this pool that have small reduced cost based on the dual multipliers of the optimum solution. Moreover, we find additional columns for the pool by applying small mutations. Finally, we let the ILP run for this pool of rosters for 15 minutes. In the resulting solution almost all occupancy constraints are met (sometimes one weekend night-shift needs one more nurse) with good rosters (most of the soft preferences were fulfilled, but sometimes there was a single day off).

We then implemented a set of heuristics for repairing roster problems, which also can be used to resolve new requests. The philosophy behind these was to make small changes, where each change would resolve an 'important' problem at the expense of creating an 'unimportant' problem. In this way, we can also resolve requests like people who should cooperate regularly. Furthermore, in this way we can determine consecutive 6-weeks rosters independently, which are then glued together.

References

[1] F. He and R. Qu (2012). A constraint programming based column generation approach to nurse rostering problems. *Computers & Operations Research, 39*, pp. 3331–3343.

[2] H. Hoogeveen and E. Penninkx (2007). Finding near-optimal rosters using column generation. Technical Report UU-CS-2007-002, Department of Information and Computing Sciences, Utrecht University. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.81.8622&rep=rep1&type=pdf