Making good rosters for the security personnel

Han Hoogeveen, Eelko Penninkx

Institute for Information and Computing Sciences,
Universiteit Utrecht, P.O. Box 80089, 3508 TB Utrecht, The Netherlands.
e-mail: slam@cs.uu.nl, penninkx@cs.uu.nl

1 Problem description

We look at the problem of finding good rosters for the security personnel of the academic hospital Utrecht Medisch Centrum (UMC) in Utrecht. The security personnel have to man several posts in the hospital, seven days a week, 24 hours a day. Therefore, the people work in three shifts: Morning, Late, and Night; the required minimum number of persons varies per shift. Except for a number of part-timers, each worker has a contract for 36 hours per week, but they are assigned approximately 34 hours per week on average to compensate beforehand for extra work that has to be carried out to replace people that are ill. Next to the ordinary working shifts, there are the so-called stand-by shifts, during which a worker can be called as a replacement. Finally, the workers have to follow a training session once-in-a-while, which is done groupwise during the Wednesday morning shift. Our task is to produce a set of good rosters for a one-year period. These rosters have to satisfy several regulations, which decree for instance that the number of hours worked per day should be reasonably balanced each week, that the number of consecutive Night shifts is at most 4, and that there should be enough time off after the last Night shift. Furthermore, there is a demand to consider rosters in which the shifts follow the order Morning-Late-Night. Finally, the rosters have to be personalized, such that they reflect the personal preferences as much as possible; we wish to maximize the total satisfaction, with the side-constraint that the unluckiest person is not extremely unlucky. Note that in the current situation there are standard rosters: in week 1, person i gets roster i, and when the week is over, he moves up to the next roster in the list, until the whole cycle of weeks has been run.

2 Solution approach

Here we discuss our initial solution approach; the problem has been changed recently (see Section 3). Initially, all 35 workers had the same function. We have measured their preferences concerning:

- The number of Morning-Late-Night shifts in a basic sequence.
- The number of days off after the last Morning, Late, and Night shift, respectively.
- A fixed weekly day-off.

E. K. Burke, H. Rudová (Eds.): PATAT 2006, pp. 437–438. ISBN 80-210-3726-1.

We have used this input to compute the quality of a year roster for each person. Here we do not add the stand-by shifts; these are added later. Our solution approach is based on integer linear programming. Assuming that we have the complete set of possible, feasible rosters available, we can model the problem of finding the optimal set as an integer linear programming problem, in which we use a binary variable x_{js} to indicate whether person j gets a roster s, or not. The constraints in this ILP enforce the minimum occupancy and the fact that we can assign only one roster per person.

Since we do not know the full set of rosters, and since we cannot handle such a big ILP problem, we solve the LP-relaxation approximately using column generation, and we use this knowledge to find an approximate solution. To get the column generation going, we enumerate all (approximately 70.000) four-week rosters; this enumeration step is necessary to deal with the constraint that the workload per week should not vary too much. Moreover, we can easily compute the satisfaction that this roster provides to each of the employees, which enables us to remove the ones that do not score satisfactorily well. We use these four-week rosters to build our one-year roster. Ideally, we would be able to solve the pricing problem by solving a shortest-path problem in a layered graph, where each node in a layer models a four-week roster. This is computationally infeasible, though, and therefore, we solve the pricing problem for only two or three four-week periods at a time: given the 'best' partial solution for the first i four-week periods, we solve the pricing problem for the periods $i + 1, i + 2$, starting with the fixed roster for period i. We then add period $i + 1$ to the partial solution, and move one. In the meantime, we compute a number of additional rosters that we believe are worthy. After we have 'solved' the LP-relaxation, we solve the original ILP for the set of generated rosters.

3 Continuation

Before it could be run in practice, the problem got changed by a function differentiation: two groups of six people were split off. For the remaining 23 people, we can still apply the approach described above, but it does not work for the groups of six. Column generation is known to work well in case there are many feasible solutions, and for the groups of six it is hard to just find a solution that satisfies the basic constraints. Therefore, we have just started a constraint satisfaction approach.